send check or money order to: People's Computer Company
P.O. Box 310
Menlo Park, Ca 94025

name_____
address_____
_____
zip

what kind of computer do you use?_____

✳ subscriptions start with 1st issue of school year

$4 for 5 issues
($5 Canada & overseas)

☐ 72-73 school year
☐ 73-74 school year

RENEW NOW

TO:

VOLUME 1 NO 5 MAY 1973
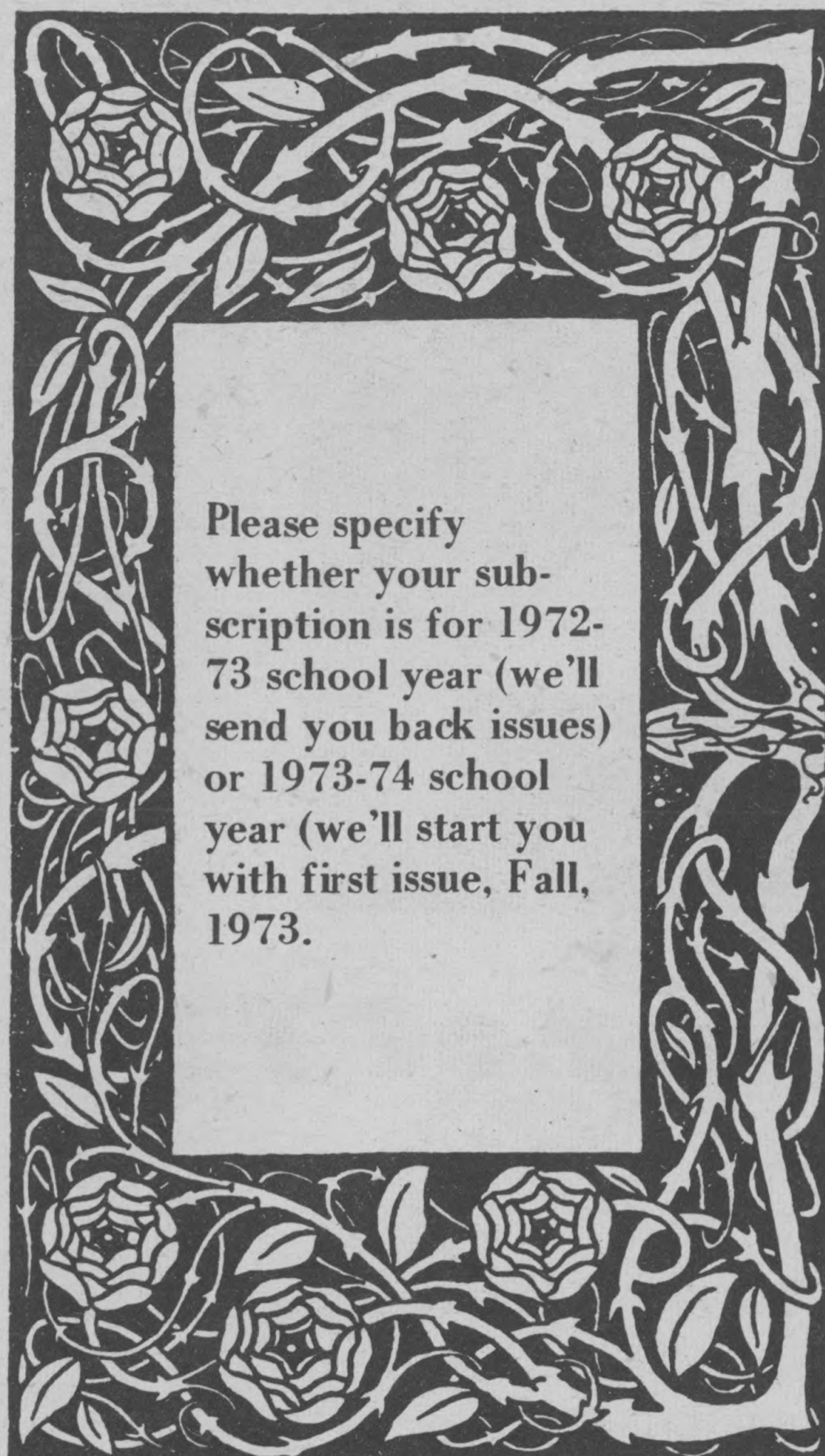
PEOPLE'S COMPUTER COMPANY

RENEW NOW
PCC 1/2 PRICE FOR STUDENTS

If you are an elementary or secondary school student, you can subscribe to PCC for ½ price ($2 for 5 issues). Send us some evidence that you are a student — for example, a letter from a teacher on your school's letterhead is O.K.

Please specify whether your subscription is for 1972-73 school year (we'll send you back issues) or 1973-74 school year (we'll start you with first issue, Fall, 1973.

PEOPLE'S COMPUTER COMPANY is published 5 times during the school year. Subscriptions begin with the first issue in the Fall.

Single subscriptions — $4 for 5 issues [$5 outside U.S.A.]. Group subscriptions, mailed all to the same address —

10 or more	$3.00 each
30 or more	$2.50 each
100 or more	$2.00 each

Send check or money order to: **PEOPLE'S COMPUTER COMPANY, P.O. Box 310, Menlo Park, Ca. 94025**
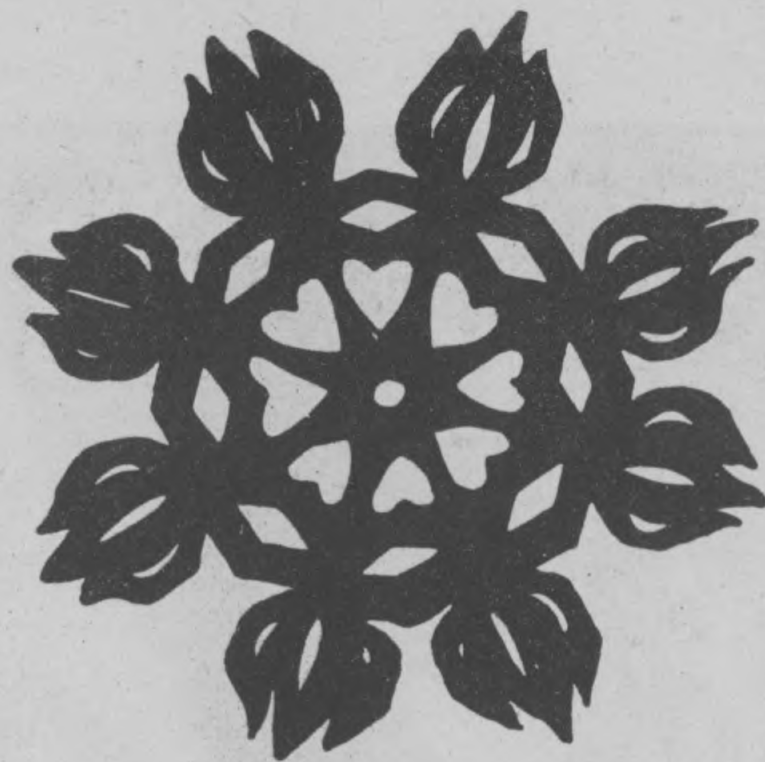
name _____

address _____ zip _____

school _____

school address _____ zip _____

☐ I am subscribing for 1972—73 school year. Please send back issues.

☐ I am subscribing for 1973—74 school year. Start my subscription with first issue in Fall, 1973.

SCHOOL COMPUTER SERVICE CORPORATION

Here is another sharing thing — schools getting together to share a computer. Any more out there?

ba

This non-profit organization was incorporated in North Carolina in August 1970. Six School Superintendents comprise the Board of Directors

The Corporation is purchasing a GE265 Time-Sharing Computer System. It is making computing available to schools at minimal cost and is providing a unique fraternity among its users.

There are currently seven full-member school systems sharing the purchase cost. Two of these school systems each have agreements with two other school systems to share their financial commitment to the Corporation. Three other school systems or schools are buying time on an annual basis.

Each school system is responsible for its own telephonic communication with the computer and each full-member may have 8 terminals on-line simultaneously. Currently there are 38 on-line terminals linked to the system and the largest number that have been on-line at any one time, to date, is 21.

The computer system is used primarily for instructional purposes, although each full-member is allowed 15 hours per month for administrative work (in the batch processing mode). The administrative work is to be done when the system is not required for instruction — nights, weekends, vacations.

Further information may be obtained from

Edward G. Blakeway
School Computer Service Corporation
102 Chamberlain Street
Raleigh, North Carolina 27607

odyssey

schedule

PC Center Summer of 73

We will probably do things like this —

- Do-it-yourself-hands-on-courses. Sign up for a "course." Play computer games or learn BASIC or both or ??? *6 times . . . 2 hours each time . . . $20* Arrange your own schedule. Here are the possibilities —

Tuesday	1 – 3 PM or 3 – 5 PM
Wednesday	1 – 3 PM or 3 – 5 PM
Thursday	1 – 3 PM or 3 – 5 PM

  Come once a week for 6 weeks or twice a week for 3 weeks or . . . whatever schedule *you* like.

- Or sign up for the "Pete and Ray Show" described elsewhere on this page. This is a course for people who are serious about learning how to program (in BASIC) and want more structure, more help from real live people.

- Or come to MUSIC NITE — every Wednesday evening, 7 – 9 PM. We will try to run the following schedule.

  7–7:30 Guitar Group Grope — for people who are rank beginners trying to learn folk guitar. We will lean on each other while we practice simple nitty-gritty stuff and together, build a place for beginners to learn guitar.

  7:30–8 Guitar stuff lead by Pete Sessions — for "advanced beginners." If your fingers can't keep up, maybe your voice can — sing along while the guitar pluckers play old time folk songs.

  8–9:00 Music games — on computer and off computer — for those who want to dig into music theory, composing and the like. For others, computer games for fun.

  Yes, Wednesday nite ain't free — costs you $2 for the evening.

- PCC OPEN CLASSROOM ***FREE***

  This is get to know us nite. Come on Thursday nites (7 – 9 PM) and play computer games, rap about computers in the classroom or computers for people or bring your own topic!

- Or, instead of the things mentioned above, you can just buy computer time and do your own thing. The younger you are the less you pay. From $1 per hour to $2 per hour depending on age and other variables and how we feel and . . .

Want more info? Call us. (415)323-6117

HP T/S   PDP 8/L   TTYs   BASIC

2

PCC's BASIC Programming Course

Here's your chance to learn BASIC, the most common educational computer language going. The course will meet once a week, on Tuesdays, for 10 weeks beginning

JUNE 12, 1973

We will probably offer two "sections", one in the afternoon and one in the evening, depending on how many of you sign up.

Each week we will —

- Present a few BASIC *commands* for you to experiment with.

- Show you some programming *techniques* to get your ideas flowing.

- Propose some *problems* (especially games to program) for you to work on if you wish.

- Help you to plan and write program of your own choosing.

There will be NO regular textbook, but —

- We'll Xerox our class presentations for distribution.

- We'll recommend "supplementary" study material on an individual basis. We have a good library of materials right here at PCC.

- Getting your hands on the computer is the best way to learn, anyway. We have four terminals, plugged into two different computers, for you to use.

Instructors — Pete Sessions
Ray Coop
Other PCC staff
Occasional guest appearances

How to enroll for the course — CALL or WRITE

Pete or Ray or Mary Jo
People's Computer Center
1921 Menalto Avenue
Menlo Park, CA. 94025
323-6117

Enrollment for any one section (afternoon or evening) will be limited to 10 people maximum. We'll have the details of time worked out by June 1, hopefully, and we'll let everyone know.

RSVP now! Make it easy on us . . . . One more thing, we almost forgot —

Cost — $30 each person for the series. This covers materials, computer time, and instructors' fees.

1973 ACM STUDENT COMPETITION

An opportunity to submit your original ideas on any topic related to computers and their applications.

Best papers will be published in Communications of the ACM and the authors will receive awards.

Anyone who has not received a bachelor's degree before April 1, 1973 is eligible. Letters of intent should be submitted by June 11, 1973 and manuscripts by September 11, 1973. For details see the March 1973 Communications of the ACM or write to: ACM Student Editorial Committee, Department of Computer and Communication Sciences, 2076 Frieze Building, University of Michigan, Ann Arbor, Mich. 48104

Dear P.C.C.,
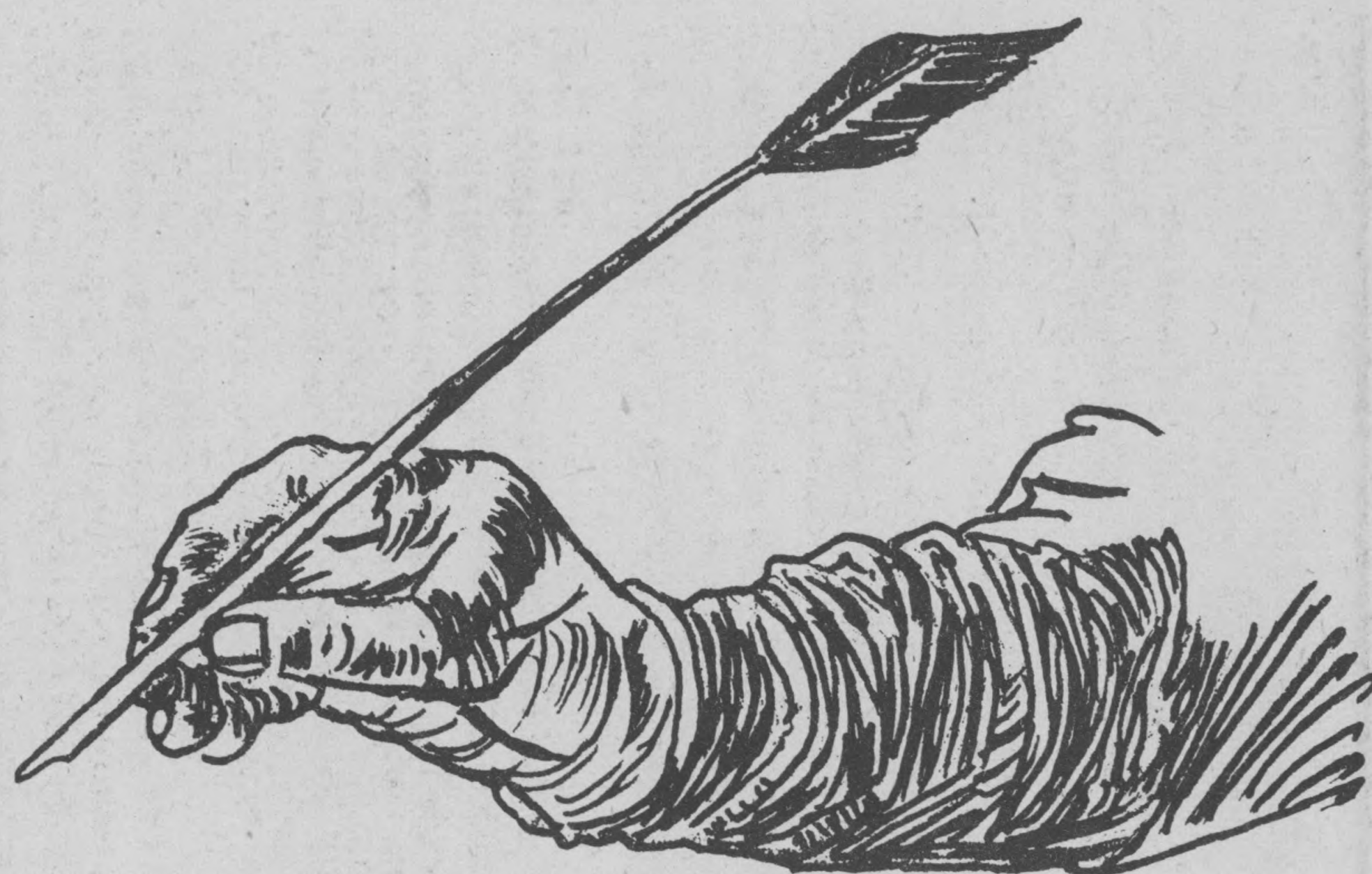
'Allo, I'm in prison in Italy. I can't stop bending me mind. 2 years to go. Please bend it some more, with yr splendid newsletter. I need recreating & educating.

Sardegna, Italy

I am a young computer professional, and the PCC and SHAFT philosophy, as expressed in the CW article is greatly to my liking. For my part, I am trying to help broaden the scope of C.S. education in my old high school by means of passing on my CW's, obtaining programs, and in the near future . . . by arranging field trips. They have a PDP-8L with two TTYs and run quad FOCAL or FOCLX. The teacher is considering BASIC.

Woods Hole, Ma.

We publish our own system newsletter called the "Punch Line" for users of our time-shared computer system. We call our system LACE for La Crosse Area Computers in Education. We would like to be able to exchange articles with you for publication. In particular, we would like permission from you to reprint yours with appropriate credit given. Please let us know of your thoughts on this matter.

University of Wisconsin
La Crosse Computer Center

Noticed the People's Computer Center in Stewart Brand's SPACEWAR article. I am part of a very far out number, also in of all places Miami. Into light and sound systems; we have created a room which we call a LUMONICS environment where we do a concert on Saturday evenings at 10 PM. Most definitely an energy center; we're in process of automating it as well as filming (videotape if money permits) when economics are in allignment. If you ever space into Fla. check us out. Miami is ripe for tapping!! Keep up the good program.

Miami, Florida

Mel Tanners'
LUMONICS
355 N.E. 59th Terrace
Miami, Florida

PEOPLE'S COMPUTER COMPANY sounds like a DP-oriented WHOLE EARTH CATALOG. I'm interested.

Dover, Delaware

Roughly 5 minutes ago, I came across a brief mention of your People's Computer Center in Saturday Review. Although the article says very little, I suspect that you are doing something similar to what I've proposed to do in my community (a small, "tight" heterogeneous district of L.A.).

Needless to say, I am excited to hear of your project — both because my colleagues seem incapable of conceiving of anything more managable or people oriented than a Billion Dollar Computer Utility (although I hope to have a few terminals beginning this summer), and because I would like to learn of your experience.

Los Angeles, California

* *

I have an Interdata Model 4 Computer, with TTY. I would like to find other Interdata users with a like machine who would be willing to exchange hardware, application programs, tapes, information, etc. I am primarily interested in educational data processing. So if any readers have developed any interesting programs, and would like to share them with this reader, please contact:

Dr. Gerald A. Silver
Asst. Prof. Business Administration
12444 Victory Blvd., No. 405A
No. Hollywood, Ca. 91606

* *

Love your paper but I must once again take exception to your cover cartoons. You keep insisting (and I agree) that computers are to serve people and not the other way around. But yet we see on your April cover two examples of machines being used to dehumanize large numbers of people.

One important use of computers which is just being recognized is their ability to humanize (!!) through individualization. One example of this can be found in the Apr. issue of "Popular Science" p. 106 on which we find a computer being used to produce paint-by-number canvasses from a customer supplied picture. I'm sure you have heard of computers being used to tailor cut suits and of course education can be made quite personal through the use of computer assisted instruction.

Such cartoons as the first, lead to a subconcious stereotyping of the form "All scientists are the same" or "All women are to have families with 3 (!!ZPG wouldn't appreciate that) children."

As for the second one, if we allow ourselves to be placed in cubicles with the computer at the center we can probably expect to turn into nothing more than tapes and this cartoon might imply this but at the same time, the cartoon is also implying that CAI can take only the form of cubicles all linked to the same speed of instruction and the same material.

I really enjoyed the cartoon on page 2 and it got me to thinking about the fact that PCC has very little in the way of flowcharting, documentation, etc. i.e., the paper work!!

Flowcharting can clear up a lot of programming problems if handled properly and I wonder whether James McCormack's son (bottom right hand corner of p. 10) flowcharts. I'm almost willing to wager he doesn't.

PILOT 73 looks really good. Now if I could just get my hands on an interactive language!! We've got a UNIVAC 1108 but it's all batch mode.

Glad to see (on p. 11) that "Down with FORTRAN" is no longer a public sentiment of yours.

How does one "do a page" for you? Size? etc.? I might try one on "how a computer sort of goes about its business" as G. Harland of Univ. of Alberta asks for on p. 10. I was thinking of CARDIAC by Bell Telephone Labs — ever heard of it?

Keep up the non-linear style. It's really attractive!   R. Granhold

Chicago, Illinois

3

LOST IN THE
CAVES

WELCOME TO THE CAVES

CAVES 1 IS ONE OF A SERIES OF GAMES THAT LET YOU
EXPLORE TREE STRUCTURES AND NETWORKS REPRESENTED
AS "CAVES." A RUN OF CAVES IS SHOWN BELOW.

```
RUN
CAVES1

WELCOME TO THE CAVES

IS THIS YOUR FIRST VISIT (1=YES, 0=NO)?0

HOW HARD SHOULD I MAKE THE CAVES?
    1=USUAL,  2=HARDER,  3=!!!  ?2

DO YOU WANT AN INTRODUCTION (1=YES, 0=NO)?1

    IMAGINE YOURSELF AN EXPLORER OF THE FAMOUS
DUZZLEDORF CAVES. YOU'VE BEEN UNDERGROUND
FOR DAYS, TRIPPING THROUGH THE CAVERNS AND
TUNNELS. UNFORTUNATELY, YOU'RE LOST, AND
YOUR FOOD HAS RUN OUT.

    THERE IS ONLY ONE PATH OUT. SEE IF YOU
CAN FIND IT.

    WHEN I TYPE A '?', YOU GIVE ME THE NUMBER
OF THE CAVERN YOU WANT TO GO TO. LIKE THIS:

WHERE NEXT?  7

ADVICE: MAKE A MAP AS YOU GO - IN THE HARDER CAVES
        YOU SOMETIMES HAVE TO GO BACK AND TRY ANOTHER
        WAY. GOOD LUCK!

YOU'RE IN CAVERN # 1
# 2    # 3    # 4     ARE WHERE YOU CAN GO
WHERE NEXT?2

YOU'RE IN CAVERN # 2
# 5    # 6    # 7    # 1     ARE WHERE YOU CAN GO
WHERE NEXT?5
DEADEND
WHERE NEXT?6
DEADEND
WHERE NEXT?7
DEADEND
WHERE NEXT?1

YOU'RE IN CAVERN # 1
# 2    # 3    # 4     ARE WHERE YOU CAN GO
WHERE NEXT?3
DEADEND
WHERE NEXT?4

YOU'RE IN CAVERN # 4
# 8    # 9    # 10    # 1     ARE WHERE YOU CAN GO
WHERE NEXT?10

YOU'RE IN CAVERN # 10
# 11   # 12   # 13   # 4     ARE WHERE YOU CAN GO
WHERE NEXT?12
DEADEND
WHERE NEXT?11
DEADEND
WHERE NEXT?13
DEADEND
WHERE NEXT?4

YOU'RE IN CAVERN # 4
# 8    # 9    # 10   # 1     ARE WHERE YOU CAN GO
WHERE NEXT?9

YOU'RE IN CAVERN # 9
# 14   # 15   # 16   # 4     ARE WHERE YOU CAN GO
WHERE NEXT?14
DEADEND
WHERE NEXT?15

YOU'RE IN CAVERN # 15
# 17   # 18   # 19   # 9     ARE WHERE YOU CAN GO
WHERE NEXT?17
DEADEND
WHERE NEXT?18
DEADEND
WHERE NEXT?19
DEADEND
WHERE NEXT?9

YOU'RE IN CAVERN # 9
# 14   # 15   # 16   # 4     ARE WHERE YOU CAN GO
WHERE NEXT?16

YOU'RE IN CAVERN # 16
# 20   # 21   # 22   # 9     ARE WHERE YOU CAN GO
WHERE NEXT?22
DEADEND
WHERE NEXT?20
DEADEND
WHERE NEXT?21

YOU'RE IN CAVERN # 21
# 23   # 24   # 25   # 16    ARE WHERE YOU CAN GO
WHERE NEXT?23
DEADEND
WHERE NEXT?24
DEADEND
WHERE NEXT?25

    !!!   SUNLIGHT   !!!

    !!!  FRESH AIR  !!!

    ...  REPORTERS  ...

WELL, AT LEAST YOU'RE OUT

AGAIN (1=YES, 0=NO)?
```
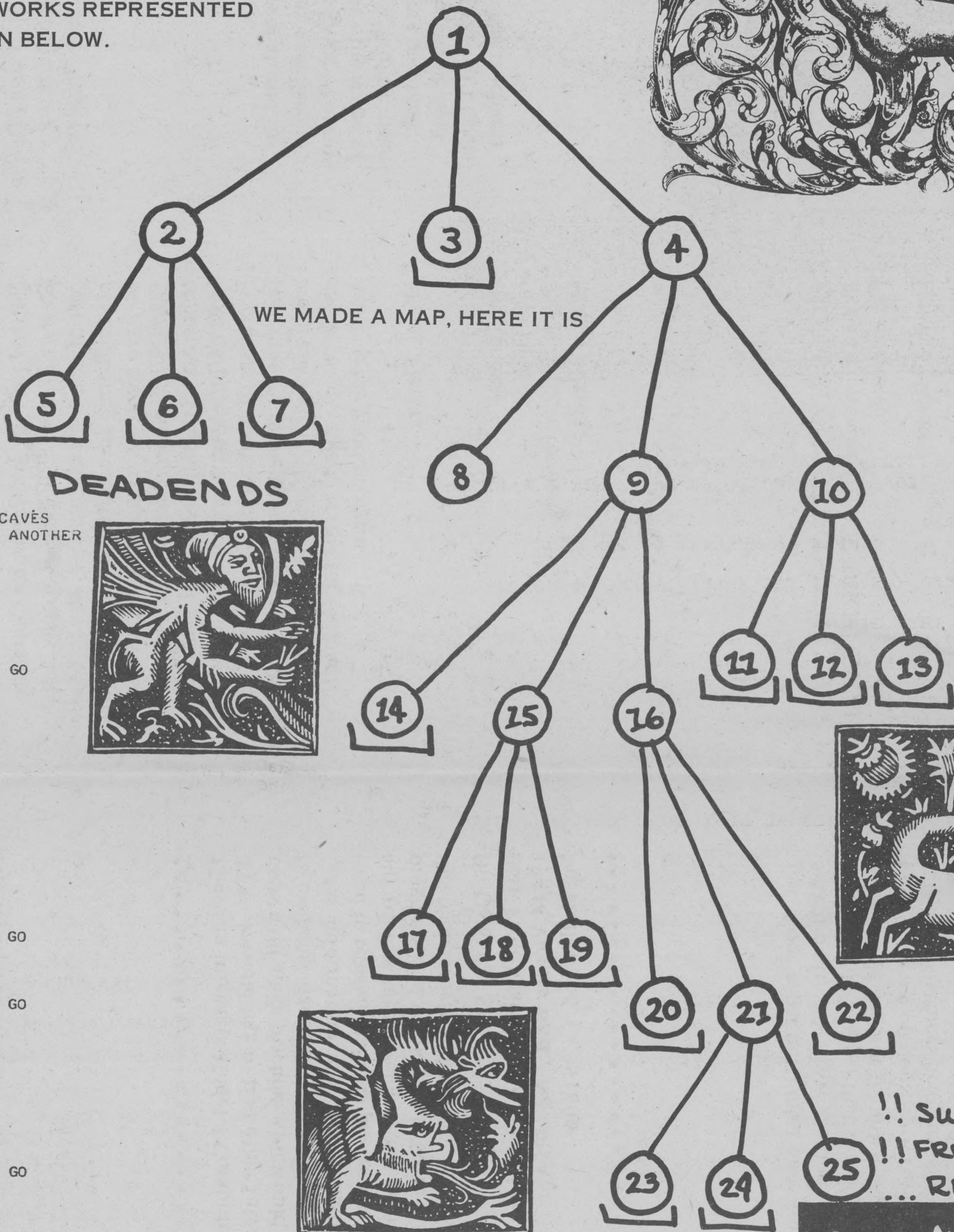
WE MADE A MAP, HERE IT IS

DEADENDS

!! SUNLIGHT !!
!! FRESH AIR !!
... REPORTERS ...

CAVES BY DAVE KAUFMAN

OTHER GAMES [SO FAR] IN
THIS SERIES INCLUDE

CAVES 2 — YOU SET UP CAVES FOR
SOMEONE ELSE TO EXPLORE

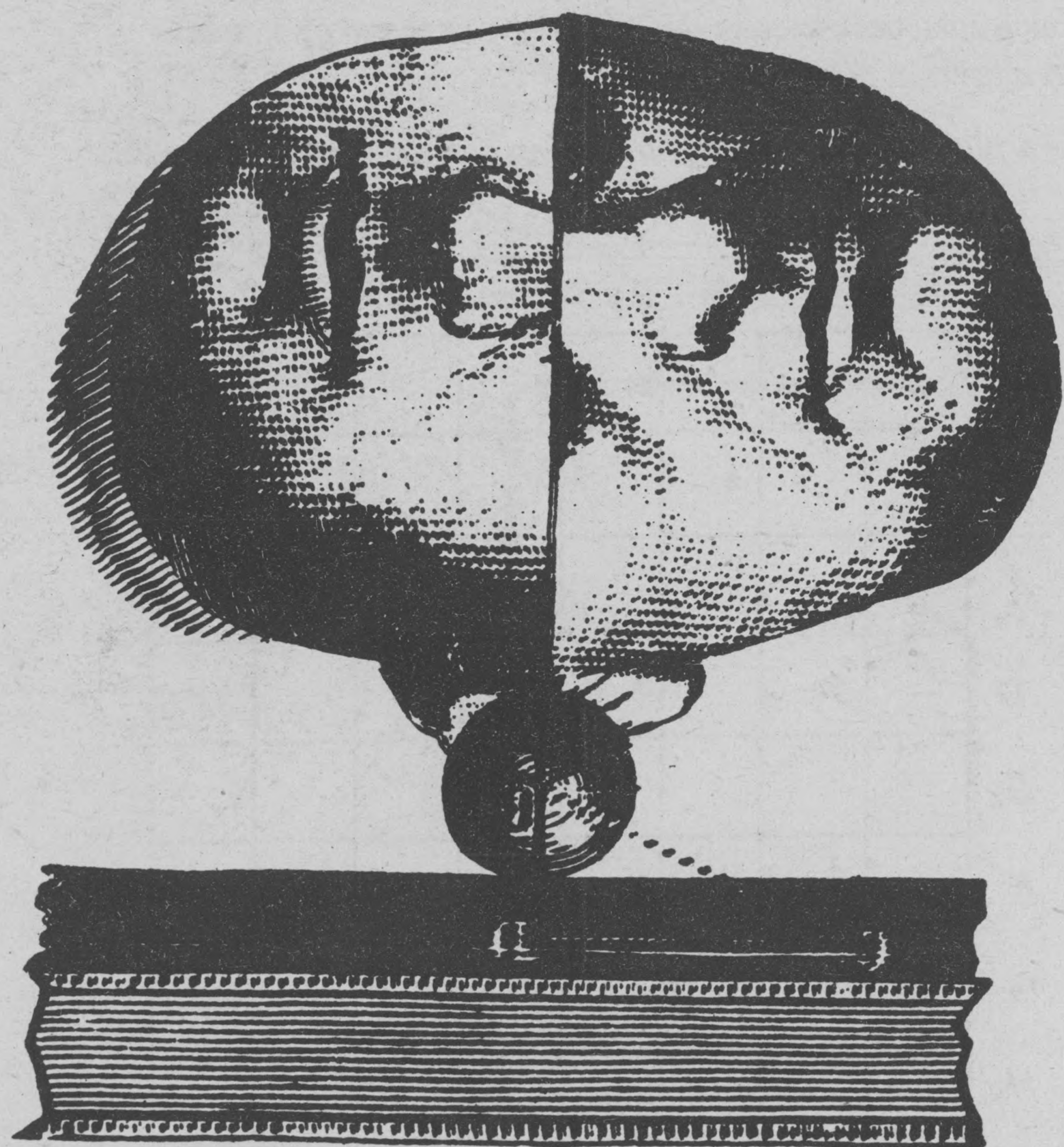CAVES 3 — SAME AS CAVES 2, BUT YOU
CAN SET UP MORE COMPLICATED
CAVES

TREES — YOU CAN MAKE CAVES, GET A
MAP PRINTED, THEN CHANGE THE
CAVES, GET A MAP PRINTED, ETC.

WUMPUS — YOU HUNT THE WUMPUS IN CAVES
WHERE YOU HAVE TO WATCH OUT
FOR SUPERBATS AND BOTTOM-
LESS PITS

↻ƎꙄЯƎVƎЯЯƎVЄЯƧƎ ↻

```
RUN
DO YOU WANT THE RULES (1=YES  0=NO)? 0

HERE WE GO ... THE LIST IS:

 3 9 7 5 8 1 4 6 2

HOW MANY SHALL I REVERSE? 6

 1 8 5 7 9 3 4 6 2

HOW MANY SHALL I REVERSE? 9

 2 6 4 3 9 7 5 8 1

HOW MANY SHALL I REVERSE? 7

 9 3 4 6 2 7 5 8 1

HOW MANY SHALL I REVERSE? 7

 5 7 2 6 4 3 9 8 1

HOW MANY SHALL I REVERSE? 3

 2 7 5 6 4 3 9 8 1

HOW MANY SHALL I REVERSE?  I give up! You take over ...
```

Would you believe a list
with 9 numbers can always
be put into order within
15 moves?


```
100 REM *** REVERSE - A GAME OF SKILL
110 REM *** PEOPLE'S COMPUTER COMPANY, MENLO PARK CA
120 RANDOM
130 DIM A(20)
140 REM *** N = NUMBER OF NUMBERS (1 THRU N)
150 LET N=9
160 PRINT "DO YOU WANT THE RULES (1=YES  0=NO)";
170 INPUT A
180 IF A=0 THEN 210
190 GOSUB 710
200 REM *** MAKE A RANDOM LIST A(1) TO A(N)
210 LET A(1)=INT((N-1)*RND(0))+2
220 FOR K=2 TO N
230 LET A(K)=INT(N*RND(0))+1
240 FOR J=1 TO K-1
250 IF A(K)=A(J) THEN 230
260 NEXT J
270 NEXT K
280 REM *** PRINT ORIGINAL LIST AND START GAME
290 PRINT
300 PRINT "HERE WE GO ... THE LIST IS:"
310 LET T=0
320 GOSUB 610
330 PRINT "HOW MANY SHALL I REVERSE";
340 INPUT R
350 IF R=0 THEN 520
360 IF R<=N THEN 390
370 PRINT "OOPS! TOO MANY - I CAN REVERSE AT MOST";N
380 GO TO 330
390 LET T=T+1
400 REM *** REVERSE R NUMBERS AND  PRINT NEW LIST
410 FOR K=1 TO INT(R/2)
420 LET Z=A(K)
430 LET A(K)=A(R-K+1)
440 LET A(R-K+1)=Z
450 NEXT K
460 GOSUB 610
470 REM *** CHECK FOR A WIN
480 FOR K=1 TO N
490 IF A(K)<>K THEN 330
500 NEXT K
510 PRINT "YOU WON IN";T;"MOVES!!!"
520 PRINT
530 PRINT "AGAIN (1=YES  0=NO)";
540 INPUT A
550 IF A<>0 THEN 210
560 STOP
600 REM *** SUBROUTINE *** PRINT LIST A(1) TO A(N)
610 PRINT
620 FOR K=1 TO N
630 PRINT A(K);
640 NEXT K
650 PRINT
660 PRINT
670 RETURN
700 REM *** SUBROUTINE *** PRINT THE RULES
710 PRINT
720 PRINT "THIS IS THE GAME OF 'REVERSE'. TO WIN, ALL YOU HAVE"
730 PRINT "TO DO IS ARRANGE THE LIST OF NUMBERS (1 THROUGH";N;")"
740 PRINT "IN NUMERICAL ORDER FROM LEFT TO RIGHT. TO MOVE, YOU"
750 PRINT "TELL ME HOW MANY NUMBERS (COUNTING FROM THE LEFT) TO"
760 PRINT "REVERSE. FOR EXAMPLE, IF THE CURRENT LIST IS:"
770 PRINT
780 PRINT " 2 3 4 5 1 6 7 8 9"
790 PRINT
800 PRINT "AND YOU REVERSE 4, THE RESULT WILL BE:"
810 PRINT
820 PRINT " 5 4 3 2 1 6 7 8 9"
830 PRINT
840 PRINT "NOW, IF YOU REVERSE 5, YOU WIN!"
850 PRINT
860 PRINT " 1 2 3 4 5 6 7 8 9"
870 PRINT
880 PRINT "NO DOUBT YOU WILL LIKE THIS GAME OF SKILL, BUT"
890 PRINT "IF YOU WANT TO QUIT, REVERSE 0 (ZERO)."
900 PRINT
910 RETURN
999 END
```

If you want a longer or shorter list,
just change the value of N in Line 150.
N can be anything from 1 to 20. The
limit of 20 is set by the DIM statement,
Line 130.

For very young people, start with N = 2
or 3, then increase N as they learn the game.
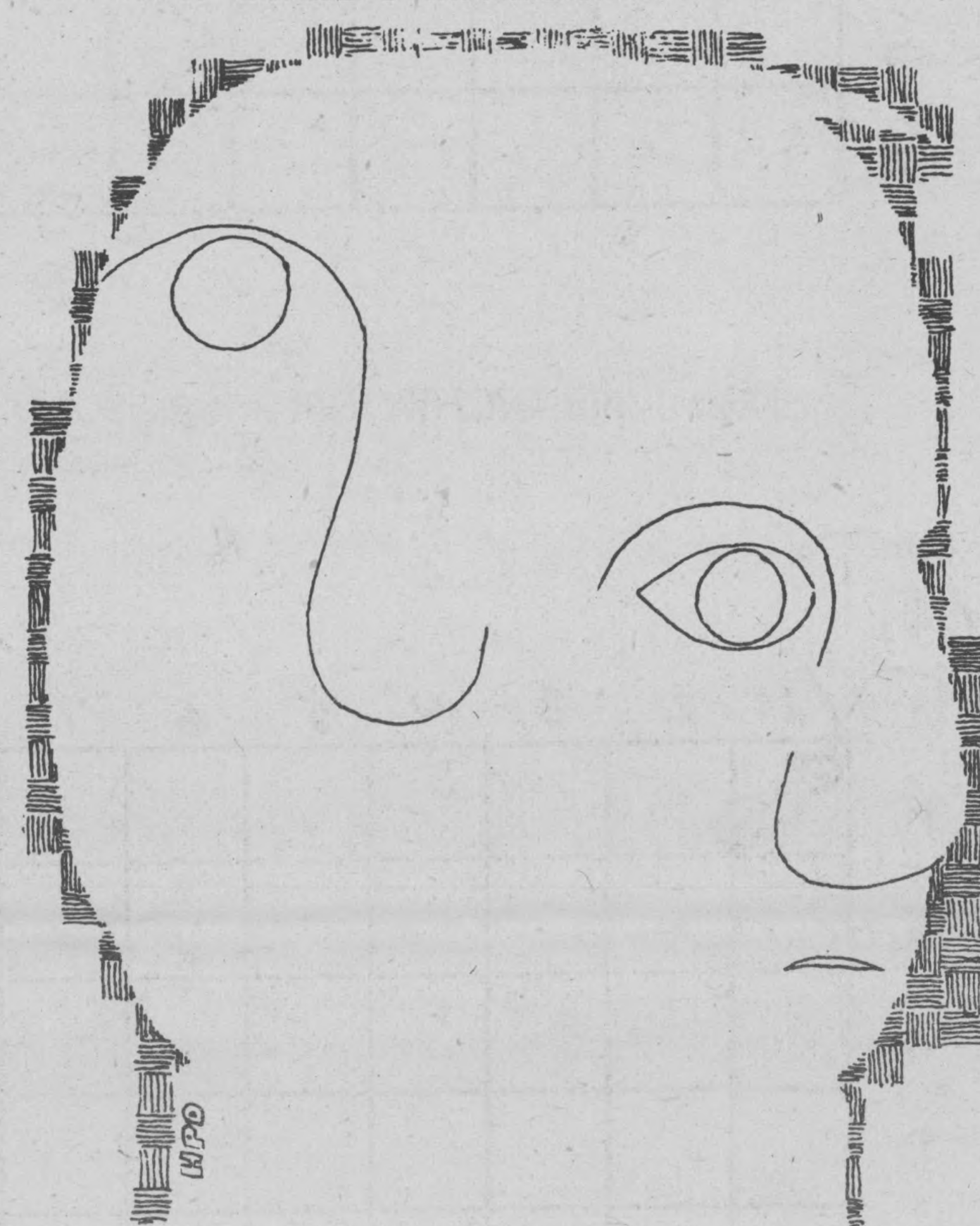
REVERSE is a simple-minded game, but dangerously addicting.

In an earlier version, we used another way of creating the randomized list of numbers, as follows:

```
210 FOR I=1 TO N
215 LET A(I)=I
220 NEXT I
225 FOR I=1 TO INT(19*RND(0))+19
230 LET X=INT(N*RND(0))+1
235 LET Y=INT(N*RND(0))+1
240 LET Z=A(X)
250 LET A(X)=A(Y)
260 LET A(Y)=Z
270 NEXT I
```

There are many ways to beat the game, but approaches tend to be either algorithmic or heuristic. The game thus offers the player a chance to play with these concepts in a practical (rather than thoretical) context.

An *algorithmic* approach guarantees a solution in a predictable number of moves, given the number of items in the list. For example, one method we found (invented?) guarantees a solution in 2N − 3 moves when the list contains N numbers. The essence of an algorithmic approach is that you know in advance what your next move will be. We could easily program a computer to do this.

A *heuristic* approach takes advantage of "partial orderings" in the list at any moment. Using this type of approach, your next move is dependent on the way the list currently appears. This way of solving the problem does not guarantee a solution in a predictable number of moves, but if you are lucky and clever you may come out ahead of the algorithmic solutions. We could not so easily program this method.

In practice, many players adopt a "mixed" strategy, with both algorithmic and heuristic features. Is this better than either "pure" strategy?

INCHWORM

BA

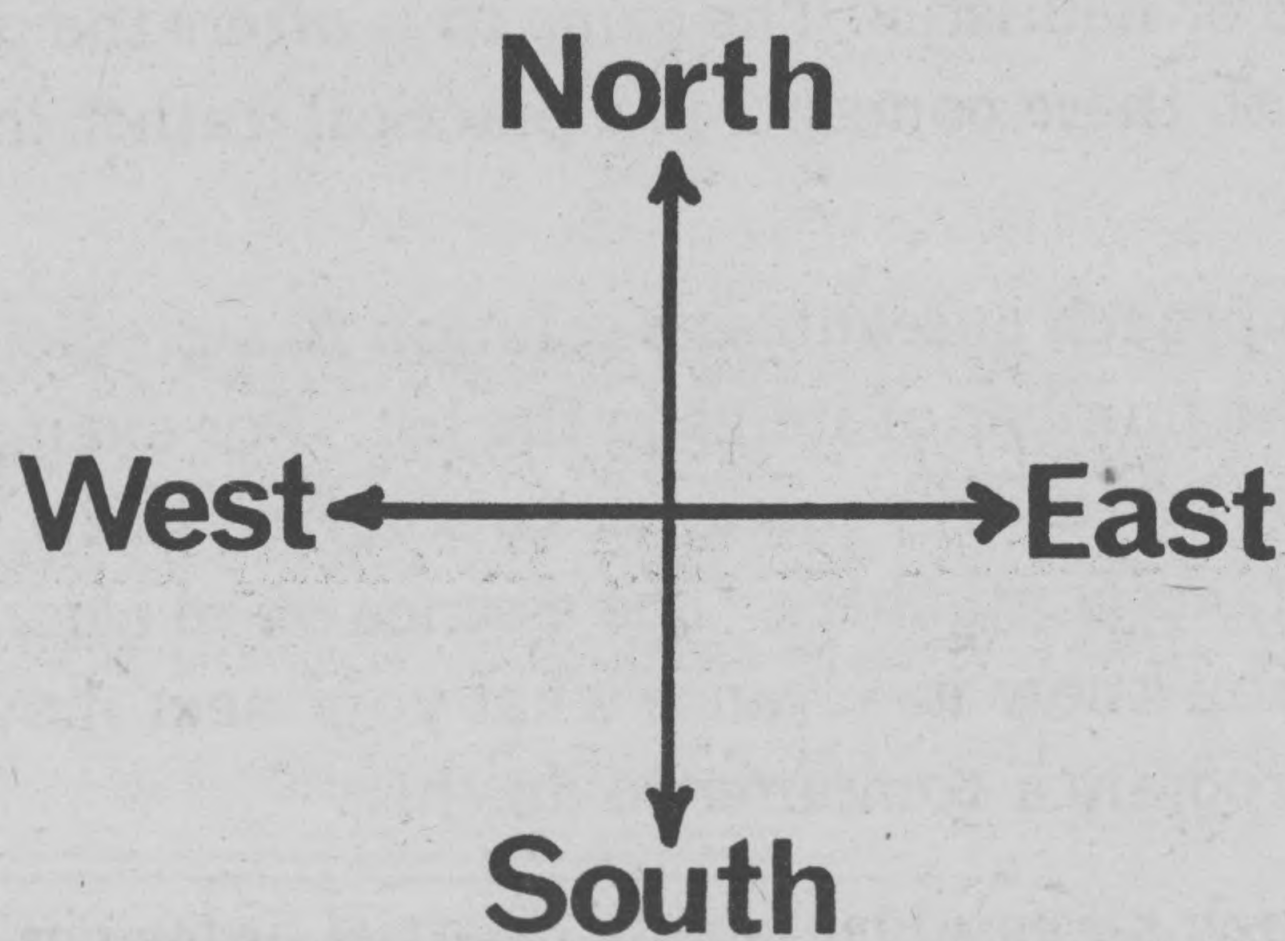Start with a "board" like the one below.

Next — put INCHWORM in square A1.

But our INCHWORM isn't just any old everyday run-of-the-mill inchworm. He is computerized! You can program him to move around the board under your control.

Here are the things he can do. He can move — one square at a time — NORTH or EAST or SOUTH or WEST.

⦿ To tell him to move one square NORTH, write N
⦿ To tell him to move one square EAST, write E
⦿ To tell him to move one square SOUTH, write S
⦿ To tell him to move one square WEST, write W

6

Suppose INCHWORM is now at square A1 and we want him to move to square C5. We could tell him like this: EEEESS

This is a program.
The program consists of 6 instructions.
Each instruction tells INCHWORM to move one square in a certain direction.

Here is a picture of the path he took in getting from A1 to C5.

Remember . . .

We started with a problem: Tell INCHWORM to go from A1 to C5.

We wrote a program: EEEESS

Then INCHWORM did what our program told him to do and moved from A1 to C5.

Below is a different program to move INCHWORM from A1 to C5.

PROGRAM: EESSSSEEENNW

Your turn: Show the path taken by INCHWORM in getting from A1 to C5 (according to our latest program.)

Somebody wrote the following program to get INCHWORM from A1 to C5.

PROGRAM: SSSSSEENNEE

Does this program work? If INCHWORM does what he is told by the program, does he end up at C5? If the program doesn't work, can you fix it?

A program that doesn't work is said to have a "bug" in it. The process of finding and fixing a bug is called "debugging."

OK, let's try another problem. We want INCHWORM to start at A1 and then move on a path that makes a box and end up back at A1. We want each side of the box to be 3 squares.

PROGRAM: EESSWWNN

Here is his path.

Somebody wrote the following program to tell INCHWORM to make the box above. It has a terrible bug which causes INCHWORM to fall off the board.

PROGRAM: SSEENNNWW

Here is what happened to poor old INCHWORM
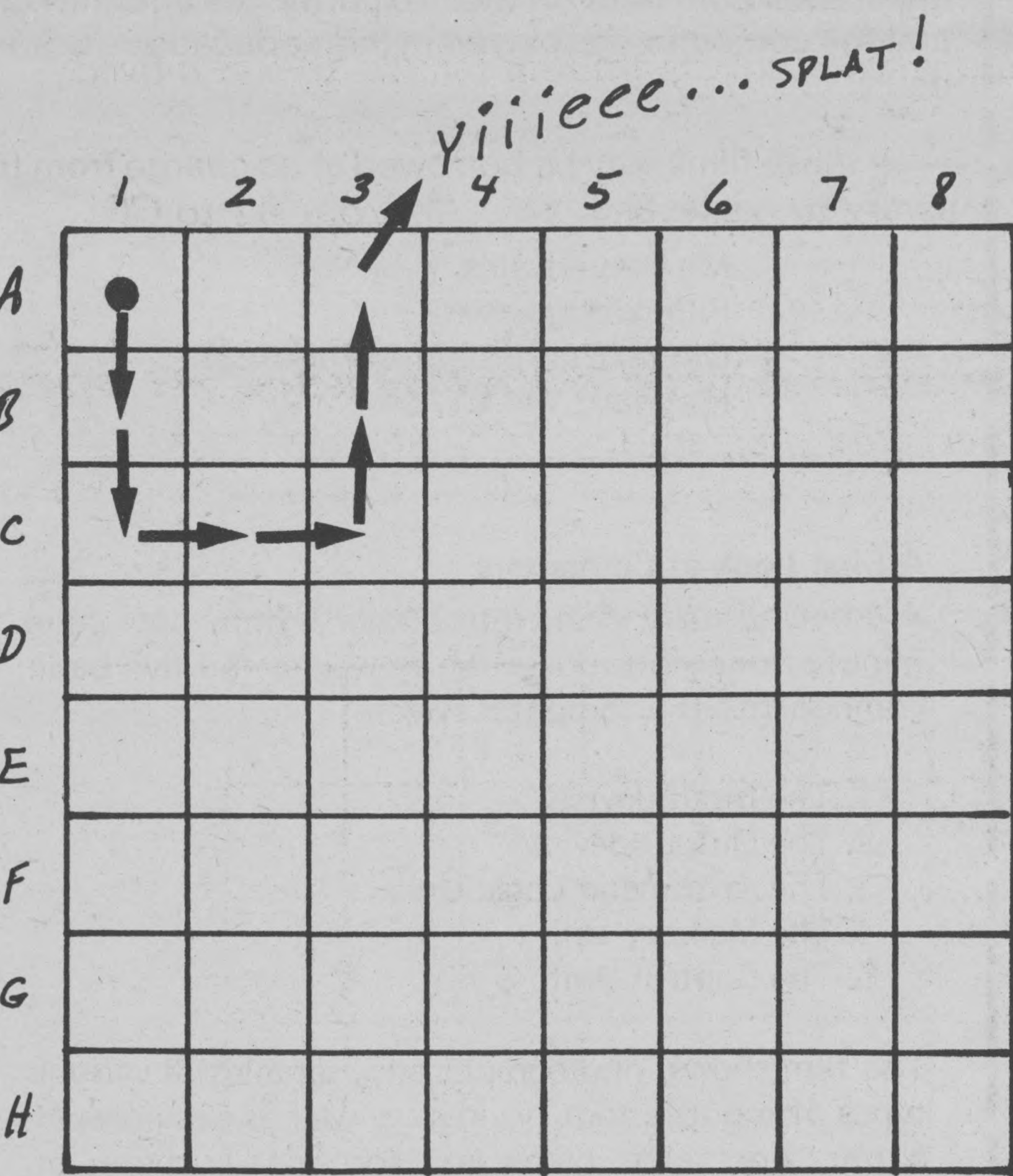
Can you debug the program? (That means can you fix it?)

Another kind of "bug" — what is wrong with the following program to make a box?

PROGRAM: EESSZEENN

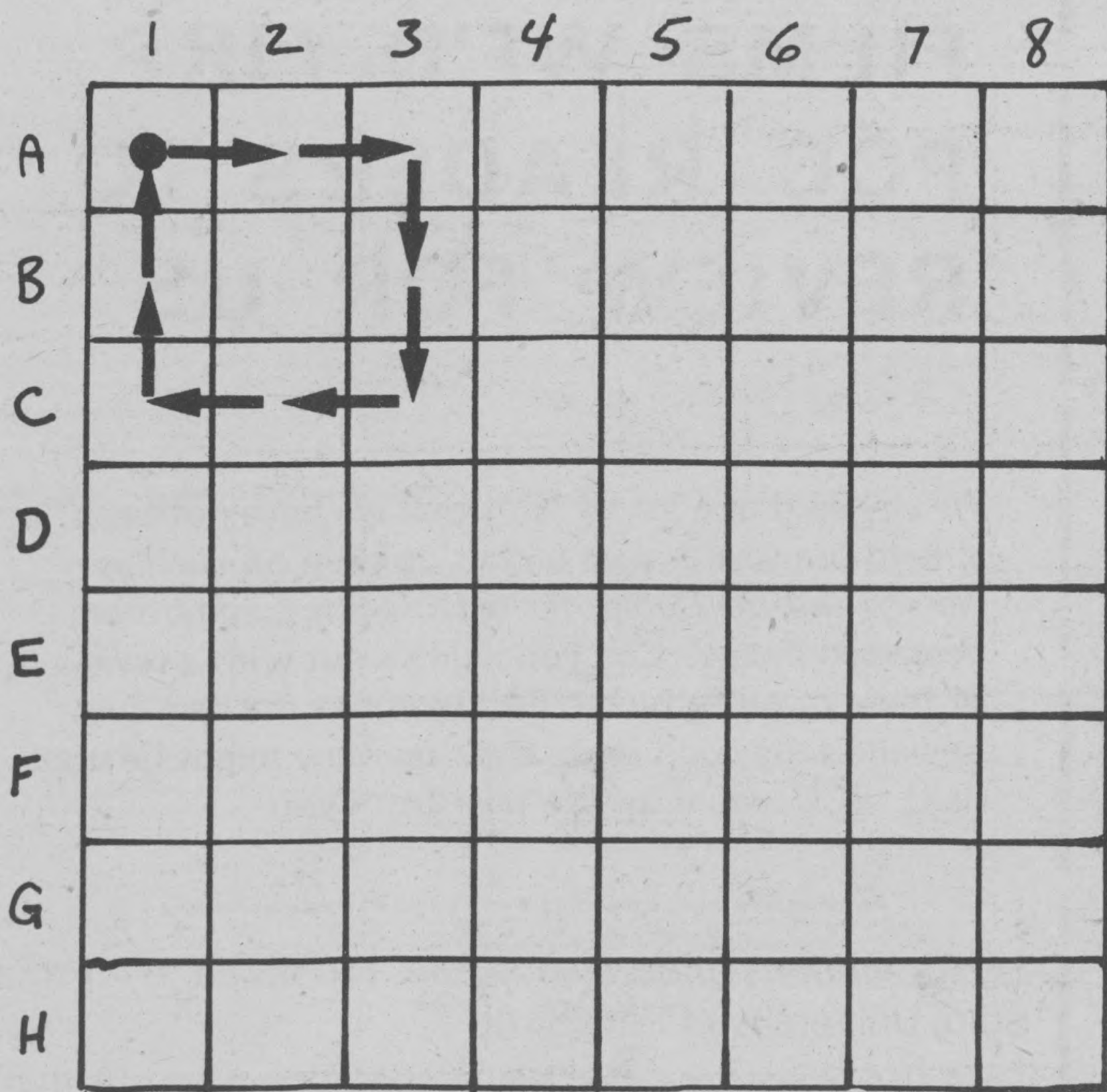If you ask INCHWORM to carry out this program, he might look it over and say (in inchworm talk, of course), "I can't do this program. I don't know how to do a Z."

Well, the moral of the story is: You can only tell INCHWORM to do the things he knows how to do, like N and E and S and W.

THINGS TO TRY

◉ Forbidden Squares. Write a program to move INCHWORM from A1 to H8 without passing through a forbidden square.

This is a forbidden square
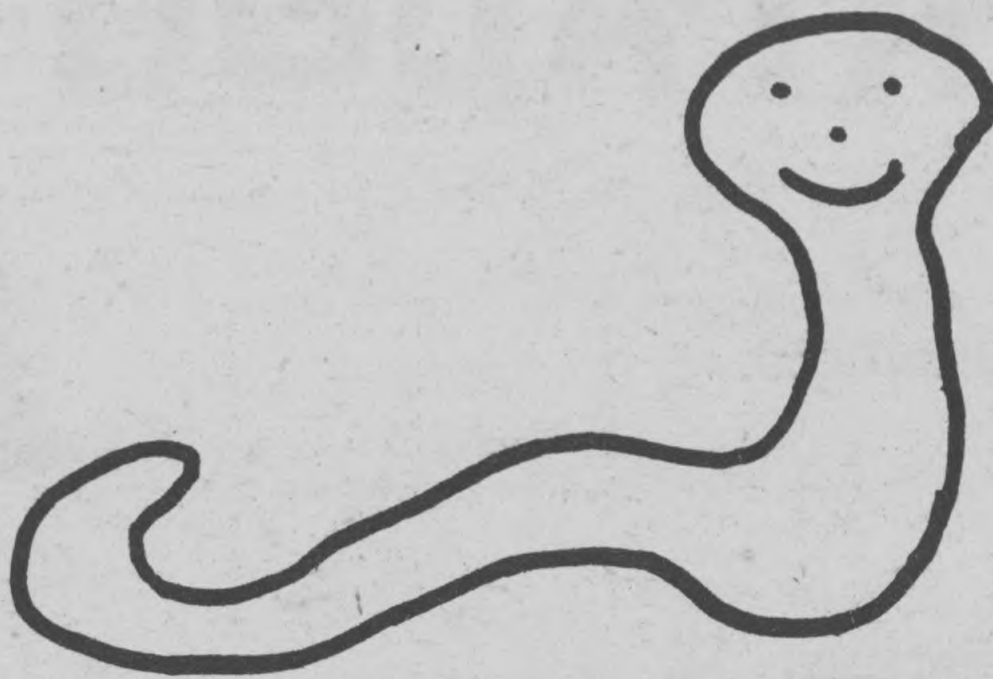
Or make a complicated maze and have a friend write a program to move INCHWORM through the maze.
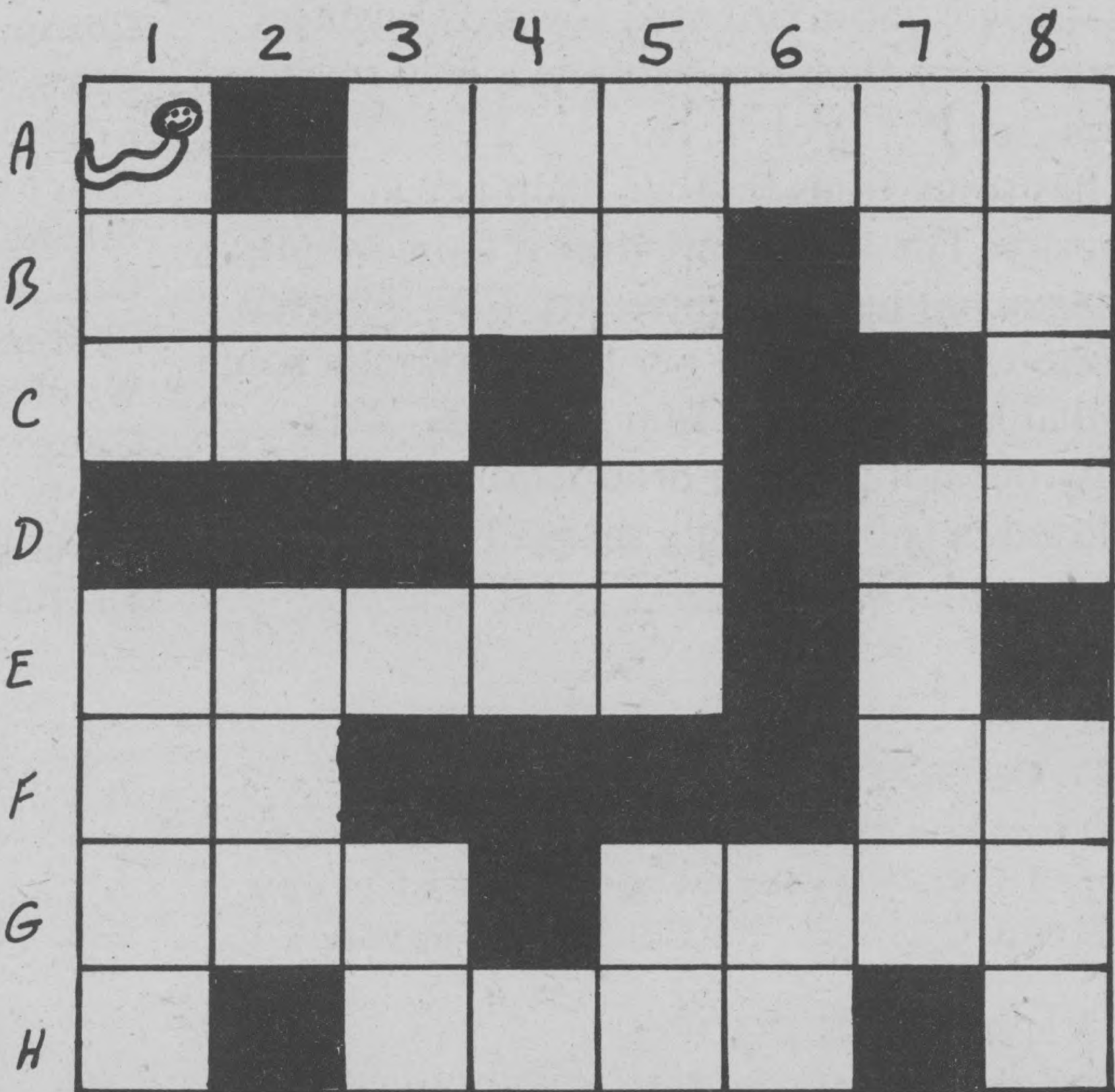
◉ Put a "random digit" in each square.

	1	2	3	4	5	6	7	8
A	2	5	6	4	3	3	8	0
B	4	3	0	1	7	7	9	4
C	9	6	8	4	9	0	4	5
D	2	2	1	7	5	2	3	0
E	4	6	0	5	1	0	2	9
F	3	2	1	2	5	3	2	1
G	5	4	9	8	3	7	5	3
H	5	0	9	1	3	4	0	7

◉ Write a program to move INCHWORM from A1 to H8 so that the sum of the numbers in the squares in his path is equal to 64. (Including the numbers in A1 and H8.) How many different programs can you write? Also try it for a number other than 64.

◉ Write a program to move INCHWORM from A1 to H8 so that the sum of the numbers in the squares in his path is equal to a prime number. Remember to include the numbers in A1 and H8.

This is just the beginning of the INCHWORM story. Be sure to tune in this Fall when we give INCHWORM a bucket of paint and a paint brush.

7

AUDIO-VISUAL ~ACCESS AND~ UTILIZZZATION

COMPUTER FILM CATALOG

Systems Film Catalog
Association for Systems Management
24587 Bagley Road
Cleveland, Ohio 44138

60 pages, spiral binding, 251 listings, 10 classification headings, alphabetical index, list of film sources, published 1972. $2.00

This catalog is more oriented towards business data processing than the two previously reviewed catalogs (see PCC Vol. 1, No. 3 — Feb. '73, p. 5), but it has some films that we hadn't seen listed previously. The listings are drawn from twenty sources, including nine university film libraries. Only one source is listed per film, although some are available from many film libraries. This catalog does not give the production date of the films listed, a short-coming shared by many film lists. Examples of listings—

CONTENTS

RIGHT OF PRIVACY
Picture takes the stand that the proposed National Data Center is a danger to every American Citizen. (NET Film Service)

Color, 59 minutes, $16.00

Source: UC

ENGINE AT THE DOOR
Discusses question of will machines ever run man. Points out that the wise and beneficial use of science and technology is man's responsibility. (NET Film Service)

B & W, 29 minutes, $10.00

Source: UC

EXPERIMENT IN MOTION GRAPHICS
John Whitney narrates this explanation of how he worked with a computer engineer and a computer to create an infinite variety of abstract film designs. Completed footage, a substantial portion of film similar to Whitney's beautiful PERMUTATIONS, is screened. Whitney's narration is a precise technical description and suggests new frontiers of computer use in the arts. Rented only in conjunction with PERMUTATIONS.

Color, 20 minutes $11.00

Source: UC

INFORMATION EXPLOSION
A non-technical film to show the computer's role in society. Uses filmed sequences of manufacturing processes and the man-in-in space program.

Color, 21 minutes, $7.30

Source: PSU

CLASS OF '01: COLLEGE OF TOMORROW
Explores the use of computerized and multi-media teaching systems in the universities and colleges of the future. (McGraw-Hill)

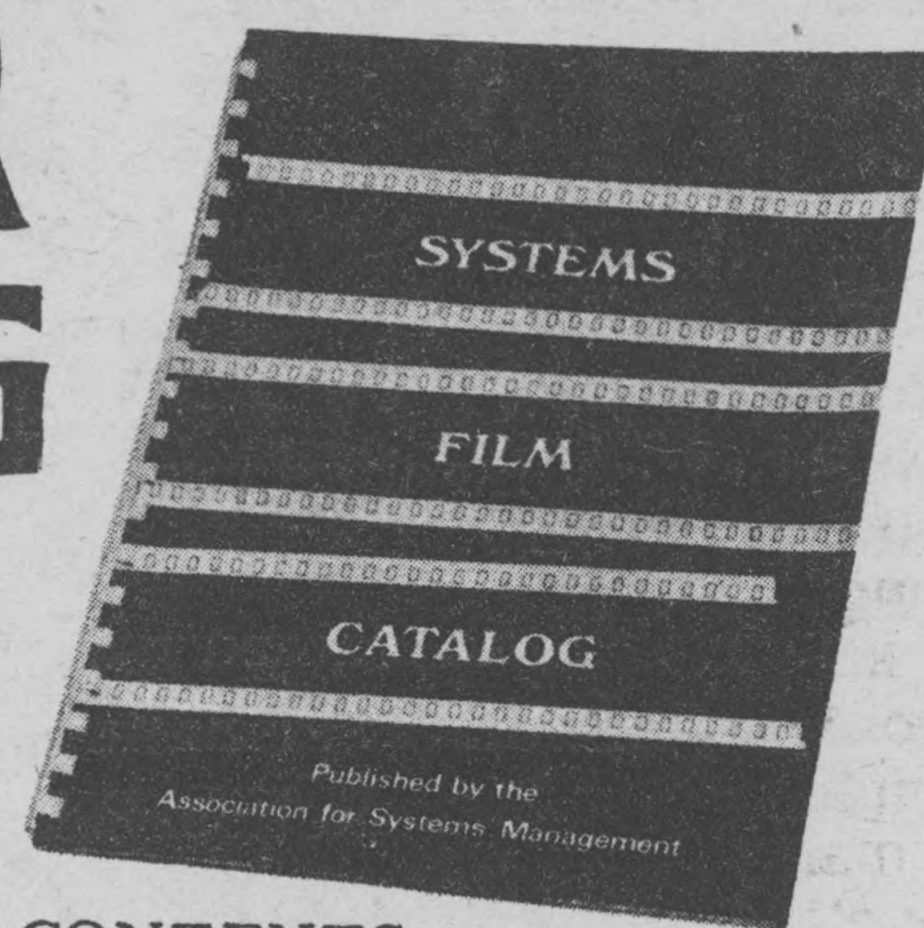Color, 25 minutes $11.50

Source: FSU

COMPUTER ANIMATION
Discusses how, as animated films have become indispensable tools for educational and training purposes, the role of computers in their production has steadily increased. Shows and discusses animations generated by both analog and digital computers for weather displays, architectural and engineering stress analysis, and aircraft navigation and landing instruction, and also as works of art.

Color, 30 minutes $23.00

Source: UC

DECISIONS, DECISIONS, DECISIONS
A description of Simulations, or "games", using the Univac I at Franklin Institute, Philadelphia, Pennsylvania. The film, made during an actual "play", summarizes the objectives, goals, and results. Tangible as well as formulative, realized by the participants. (Univac)

B & W, 12 minutes, Free

Source: UD

MARK OF MAN
A study of the ways man has used data processing from ancient times to the present. (General Dynamics)

Color, 10 minutes, Free

Source: GD
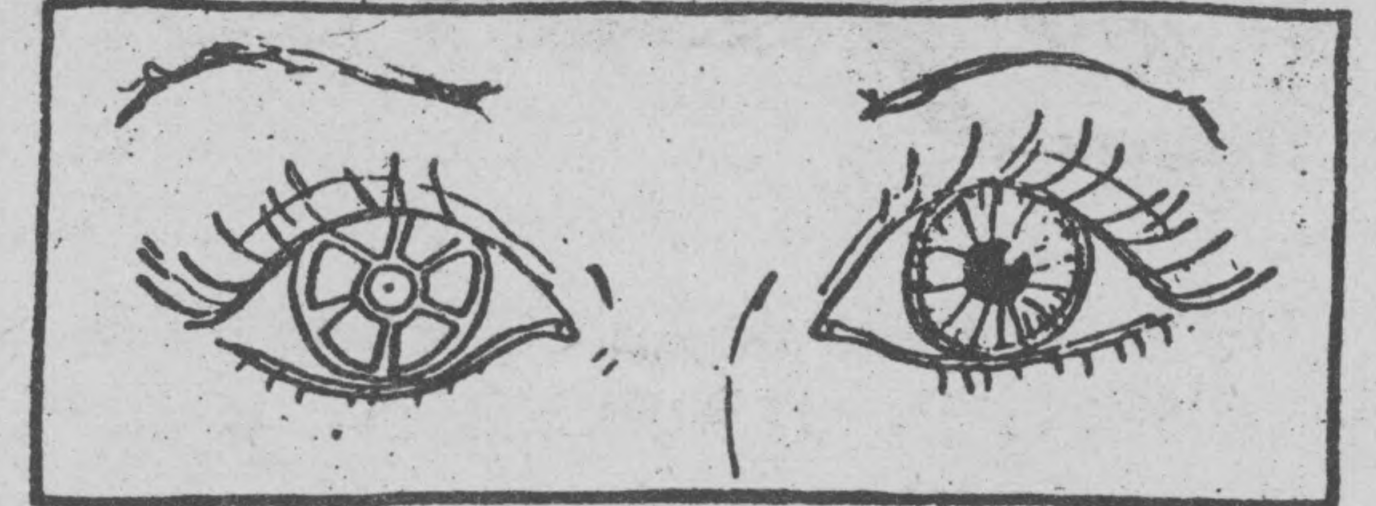
FILMS WE'D LIKE PCC READERS TO REVIEW FOR US

We've heard of a lot of films that we have not been able to preview or pass on our opinion of, such as *Neferti and the Computer* (a BBC film), and others described below. Can you help us out with a review of these or other films? Tell us about the bad ones as well as the good ones, and how they might be used used. PCC would love to hear from you!

....

My Computer Understands Me produced by Project Solo, University of Pittsburgh

This film explores the ways that the computer has strengthened thinking for students who have had the full power of computing made available to them. Statements are made concerning the effect of the computer on learning and the future directions of computers in education. *My Computer Understands Me* deals with the ways that these learning improvements reveal themselves in the real school world of today. (Color, 1 hour)

PROJECT SOLO produced by Project Solo, University of Pittsburgh

A documentary film which describes Project Solo, an experiment in regional computing in Secondary Schools. This film, which takes place right in the classroom, describes how this project uses the computer as a teaching tool. Students who have been intimately involved in the project comment on their experiences and the roles that the computer has played in their education. (Color, 1 hour)

➝ these films may be borrowed at no charge from the library by contacting:

Mr. Steve Kallis
Film Library, 6-B2
Digital Equipment Corporation
Maynard, MA 01754

....

A First Look at Computers

A brief, visually rich, introductory film—only nine minutes long—introduces the viewer to the five basic components of a computer system:

1. The Input device
2. The Output device
3. The Arithmetic Logic Unit
4. The Memory Unit
5. The Control Unit

The film shows, photographically, samples of various types of input/output devices, as well as examples of other computer hardware, for the purpose of orienting the student to the physical world of computers.

Animated sequences are used to explain the internal workings of the computer—showing component relationships and component interactions as the processing, storage, and transmission of data takes place.

The power of the computer as a tool to man is explained by emphasizing the speed, accuracy, and repetition of calculation possible once the computer has been programmed to perform a specific task.

"A First Look at Computers" will likely be of use in introductory Data Processing courses at the secondary, college and university levels.

Release Date: February, 1973
Length: 9 minutes
Purchase Price: $100.00
Preview to Purchase: $10.00

Film Order Department
Learning Resources Center
College of DuPage
22nd Street and Lambert Road
Glen Ellyn, Illinois 60137

The first annual USA-INTERNATIONAL ANIMATION FILM FESTIVAL, which took place at the New York Hilton Hotel, terminated five days of screenings and meetings with the awarding of prizes on Wednesday evening, Nov. 22, 1972, to entries from many nations.

When asked about plans for 1973, Festival Director, Fred Mintz, said, "Greatest interest was expressed in all of the new technologies that are available to the animated filmmaker and we shall therefore, underline these new technologies next year. There will be demonstrations and discussion; for example, of electronically generated visual and sound images and those who have begun to explore these domains will participate. Great emphasis will be made in computer animation and a Symposium featuring several sessions of computer animation will take place during the Festival in 1973 . . ."

The second annual USA-INTERNATIONAL ANIMATION FILM FESTIVAL will be held in New York in November of 1973. The precise dates will be announced. The articles by Ken Knowlton, Peter Foldes and Charles Csuri are from the 1972 Festival catalog.

Computer-Produced Films
By Ken Knowlton

A movie is a sequence of pictures (plus sound); each picture consists of thousands of spots of light or color arranged in an appropriate way so as to form a recognizable pattern; these spots may be produced originally on, and photographed from, a cathode ray tube, somewhat resembling a TV tube, which is controlled in turn, by a computer that sends signals specifying exactly where on the screen, and just how bright, each spot should be.

This may or may not be a good way to make a movie: it depends on how clever the programmer is who gives instructions to the computer specifying how the computer is then to give "instructions" to the cathode ray tube, and it depends on the pictorial content of the picture, i.e. on whether the desired picture may be defined in a mathematically or logically simple way. For example, it is easy to "define" a checkerboard in terms the computer can deal with; it is hard to define Mickey Mouse.

Results to date indicate that computers are indeed useful for animated films in scientific and educational areas where the subject matter and scenes depicting it have logical properties; computers are also useful for producing abstract designs and art by blending various degrees of symmetry, periodicity, randomness (from a "random number generator"), regularity, irregularity, and other factors or

Ken Knowlton is an innovator in the area of computer graphics and explorer of the art-technology interface; has collaborated with artist-filmmakers Stan Venderbeek and Lillian Schwartz. He is the author of several programing languages, among them: L⁶—a low level list processing language. BELFLIX—a general language for computer prodution of still pictures and movies. TARPS—a two-dimensional alphanumeric raster picture system for artistic stills and movies. EXPLOR—a generator of images from Explicit Patterns, Local Operations and Randomness.

features which can be given precise (though not necessarily the customary) meanings.

The future of computer animation for scientific and educational purposes is clear: it will certainly expand, and do so in proportion to effort spent on developing computer programs for specific areas, such as chemistry. Artistic usage is not as easy to predict, for many reasons: the "goal" is not as clear and not as amendable to logical analysis; human response to a particular program's output, among the millions of possible programs or combinations that might be tried, is simply not predictable until the artwork is in fact produced—because pictures defined by the logical workings of programs are not easily visualized by inspecting the programs. What is clear in "computer art" thus far, is that new graphics are resulting which never would have been done by hand.

Computer Research
By Charles Csuri

At Ohio State University an animation/graphics programming language on a mini-computer has been developed which makes the computer system natural to use and one which makes the system more powerful in terms of itself. Specific computer techniques and algorithms are programmatically combined to make complex interactive programs that are general enough to be applicable to any structure. Structures, in turn, can include any drawing or motion sequence the artist creates.

The artist is seated in a computer environment, working with electronic tools to do film animation. The drawings are displayed on the face of the computer's cathode-ray tube— a television-like screen. When the artist instructs the computer to induce movement, follow a prescribed path, or change from one shape to another, the computer responds.

The artist controls the movement of his drawings through a graphics programming language. He specifies these changes with commands and special devices. For example, with a single command he can shade 3-D objects and then adjust gray levels with dials. By simply adjusting the computer's switches and dials, the artist makes the moving drawings speed up or slow down or become larger or smaller. By pointing with a stylus, he positions or erases objects on the screen.

To supplement a visual image with explanatory words or sentences that appear on the screen, he types them in through the keyboard. The computer has stored 4,000 software characters, signs, and symbols as well as a wide variety of type faces. He controls their movement, size, and orientation as if they were pictures.

Using a special sonic pen, the artist can draw in two and three dimensions with the results displayed on the computer's screen. He might draw a helicopter on the screen, and quickly make it hover or move in any direction, its two blades turning at different speeds. He can construct a sea turtle and have it shaded by the computer while swimming through three-dimensional space in real time with twenty shaded birds and twenty butterflies flapping their wings, moving about the computer's screen —each one with independent movement and speed. When he is ready, the inherent timing processes of the computer system will help the artist record his efforts on film.

Charles Csuri
Professor of Art and Director
Computer Graphics Research Group
The Ohio State University
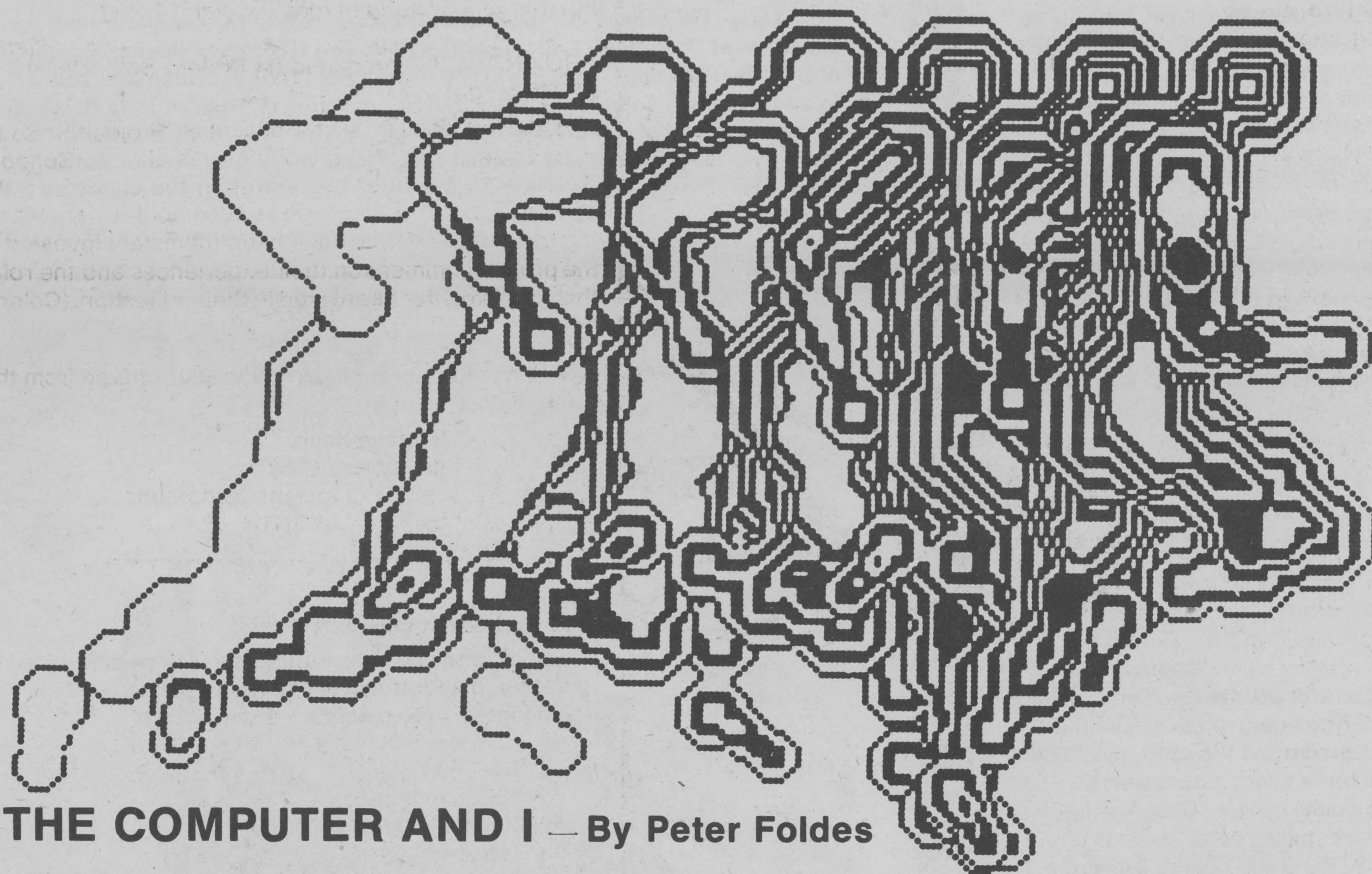
THE COMPUTER AND I — By Peter Foldes

Peter Foldes, one of the most gifted freelance animators, attached to the French Television Research Center, relates his experiences while working with Computer Animation at the National Research Council of Canada.

When I was much younger I was a painter and came to making animated films through paintings. It seemed to me inevitable that in this age of motion, transformations and continuous change, paintings have to move too. Motion pictures could be motion paintings.

So I believed that motion paintings (animated films) will replace the static canvas. That, of course, was not true; it may yet become true when the cassettes really arrive. Painting survived and animation became something else—as it should.

There are two ways to make animated films: either as a direct personal art form (the movie maker does everything—story, drawings, animation, shooting) or as a communal art form, like the big or small studios.

The lone film maker, like myself and most Europeans, suffers from the inevitable slowness of repetitive work. I cannot produce more than two to three minutes a month alone.

This is why I have recently got

interested in working directly with computers, that can take a large amount of complicated quantitative work off the creative artist's shoulders.

However, it became soon evident that computer-aided animated films will not replace the hand-made ones. The computer is able to do other things—help to create other styles, other movements than the human hand. So I firmly believe that computer-made, or rather computer-aided, animation will simply enlarge the field and produce new creative possibilities, without harming hand animation. I think the expression computer-aided films is correct, because no computer will do anything without human will. Exceptions are the computer-made films utilising random principles, which are quite outside my interest. I think the future lies in domesticating the computer for the artist, and I am trying to use it simply as a tool, asking it to perform complex, precise and exactly foreseeable, controlled tasks. At present it is probably much more difficult to make a film this way, than by hand, but it is a great challenge to the creative mind. The two will complement each other.

I am glad to present at the First

USA-IAFF "Narcissus" my film made by computer, in collaboration with the Research Center. (ORTF) This film was made with a highly sophisticated computer.

My other computer film "METADATA" was made with a digital computer at the National Research Council of Canada, utilising the programs that N. Burtnyk and M. Wein have developed there. The production is by the National Film Board of Canada.

We are developing this technique and have made tremendous progress in the production and creative possibilities. The new film is called "HUNGER" and will be completed by the end of the year. This we believe will be the first fully animated figurative film made using computer techniques. I am now working on a new script for a feature film (my second) utilising a mixture of video and computer animation with hand animation. It deals with the humorous adventures of a young girl in search of love, understanding and even some kind of a god.

Animation is a means to reach the fantastic—the dreamlike—the far out. It can do everything that cannot be made in live action. It is a field that has hardly been scratched at the surface as yet.

Computerized Steinberg: Lacy 3-D
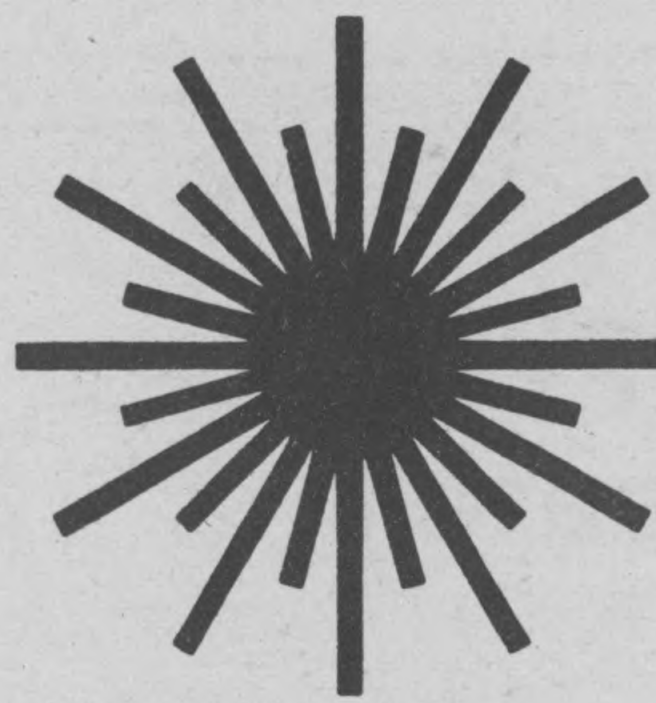
omsi

OREGON MUSEUM OF SCIENCE & INDUSTRY
4015 S.W. Canyon Road
Portland, Oregon 97221

■ *How many computers, terminals, etc.?* Two Teletypes that can be used over the phone line. A new Tektronix 4021 experimental refresh CRT for use over the phone line. A local Teletype accessing our PDP-8. A T4002 graphic computer terminal used with our PDP-8.

■ *Schedules.* When can people use the computer at OMSI? Everyday but Christmas, the computer exhibit is running in our main lobby. The really neat things for computer freaks are done in our basement and it takes a little more work to get into that. Best thing to do is to call me to see what is available.

For the last year, because we have had a lot of work to do, (thousands and thousands of mailing labels, directories and other things) as part of our production tasks . . . we have had little time for Space War and other worthwhile projects. We manipulate a data-based project which includes all math and science teachers in Oregon. We hope to get a PDP-11 sometime, but have not yet found anyone with enough money to give away (invest?). (We've got $30,000 and need another $30-40,000.)

■ *What does it cost?* Until we get a time-shared computer, we won't have any extra time to sell on the main lobby of the Museum . . . people using the computer resources in the basement are working on projects related to Museum programs and there is no charge (once they are in the programs which are supported by donors to the Museum).

■ *What do people do?* 3000 people a month visit our Public Computer Exhibit which is accessed through a DATAPOINT 3000 terminal. People come in lots of sizes, shapes and ages . . . about half the people that visit the Museum are under the age of 20.

■ *Plans for the immediate future* . . . survival. We hope we will get a handle on our production activity so we can get back to education with a small "e" We also hope to put up a new version of our public demonstration to be called DEMPAK III. It is to do a more complete job of recordkeeping and visitor performance . . . our aim is to show that it is an alternative to CAI.

■ *How can we help someone else get started doing what we are doing?*

Student Research Center — find an industry, business or museum that would let selected kids have keys to a room of their own, and a teacher or other "interface with THE SYSTEM" to take responsibility and handle legal problems. This probably won't be a public school, as schools won't take the risk of having students in them without direct supervision. Better, an art museum that doesn't mind technology activities. Frank Oppenheimer of the Exploratorium in the Bay Area is trying something like it. Other problems are the same as setting up any alternative educational project and depend essentially on how clever (and sometimes slippery) the leader is.

Other people doing it . . . in addition to the Exploratorium, the Museum of Science in St. Louis, and Sal Rositano, Ames Research Center, Moffett Field, California.

The problem — to make a Student Research Center really work, there needs to be enticing equipment and some talent . . . some of it experienced! This usually means the headquarters of a big electronics firm (like Tektronix) or a university that is not going out of existence due to the financial crunch. Much of the value of the program comes by project-oriented students feeding each other. There is a certain minimal level of experienced talent that must be added to the system, or the students get ingrown and the spirit dies. The talent would be too expensive to buy, so you have to get it donated by people who want to pass along their skills to young people. Coordination of the effort can be very tricky . . . making sure students and "experts" get together, and that the room of lab used by the group doesn't get torn apart.

■ *How to do a Software Development Group?* I'm not sure I know anymore. It seems to require a few (definitely not many) extraordinarily talented people, a lot of genuinely free computer time and an entrepreneur who wants to spread the resulting goodies around (once they are cleaned up enough for other people to make any use of them).

Rusty Whitney
Director, Computer Activities

the ChiLDReN'S MuseuM
the JamaicaWay / Boston 02130 / JA2-4800
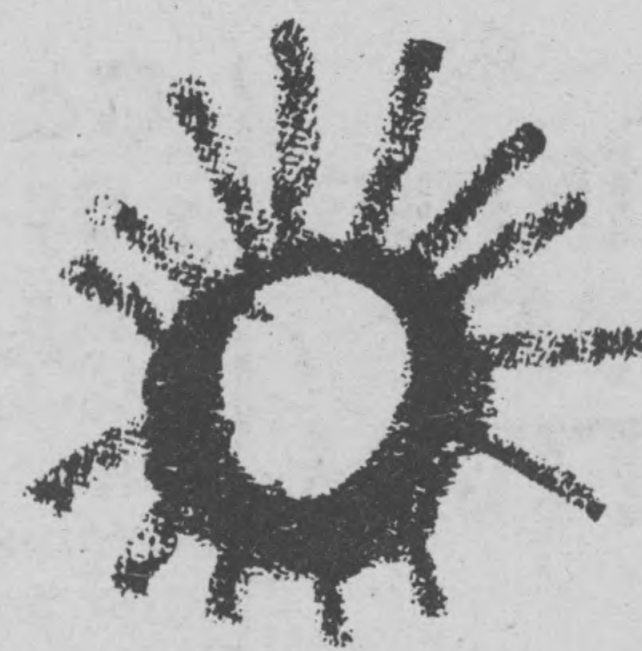
On Sunday, April 1, the Computer Center at the Boston Children's Museum celebrated its one-and-one-halfth brithday.

The occasion was greeted without fanfare . . . mostly because everybody was too busy having fun to realize the occasion existed until two days after it was over.

Since opening in October, 1971, the Children's Museum's Computer Center has developed into a place where people of all ages can come to learn about computers — how to use them, what they're like, what they can do — in an open-exploration, friendly atmosphere.

The physical setting is modest — about 600 square feet in the upper reaches of the Visitor Center, surrounded by wooden railings and stairways to other parts of the open, multi-leveled exhibit structure — as is the equipment it houses. A PDP-8I computer donated by the Digital Equipment Corporation, and acoustically coupled ASR 33 Teletype, and four old, but reliable Wang 320 electronic calculators are the backbone of the exhibit.

The PDP-8 is almost always dedicated to playing checkers with visitors via a program donated by the Polytechnic Question Society (P?S) of Brooklyn, New York. The other Teletype provides dialup access to the computer-time donations of Dartmouth College on its well-known DTSS time sharing system, of Bolt, Beranek and Newman's LOGO-supporting PDP-1 in Cambridge, and to a software development package, MIMIC, operated by Applied Data Research Corporation on First Data Corporation's DECsystem-10 in Waltham, Mass. These systems provide a vast capability for people to learn informally about computers through games, demonstration programs, and other system-supported software.

COMPUTERS ARE FOR KiDS

THE CHILDREN'S MUSEUM

Some of the things that have become most popular in the past eighteen months include a tic-tac-toe program and a Museum developed binary-search number-guessing game called YOUGUESS on DTSS, and a whole network of LOGO programs that include a simple visitor/computer conversation, games like a computerized Fanny Dooley (who loves books but can't stand reading . . . do you know why?) and the following game called MADLIBS —

```
MADLIBS

DO YOU KNOW HOW TO USE MAD LIBS? YES
DO YOU KNOW ABOUT NOUNS, VERBS, ADVERBS, AND ADJECTIVES? YES
O.K. , YOU NOUN-KNOWERS, HERE'S YOUR MAD LIB:
PICK A NUMBER BETWEEN 1 AND 2
*2
NOUN:COMPUTER
A PLACE- THE UNITED STATES
A NUMBER- 5432
A LIQUID- ORANGE JUICE
THE TITLE IS . . . . . . A NEW LAW.
IT WILL BE UNLAWFUL TO OWN A COMPUTER OR CARRY
A CONCEALED COMPUTER WITHOUT HAVING A
COMPUTER LICENSE.   THE PENALTY FOR CARRYING A
COMPUTER WILL BE THIRTY DAYS IN
THE UNITED STATES OR A FINE OF 5432 DOLLARS.
THE PENALTY IS DOUBLE IF THE PERSON IS ARRESTED
WHILE UNDER THE INFLUENCE OF ORANGE JUICE
THAT'S ALL
```
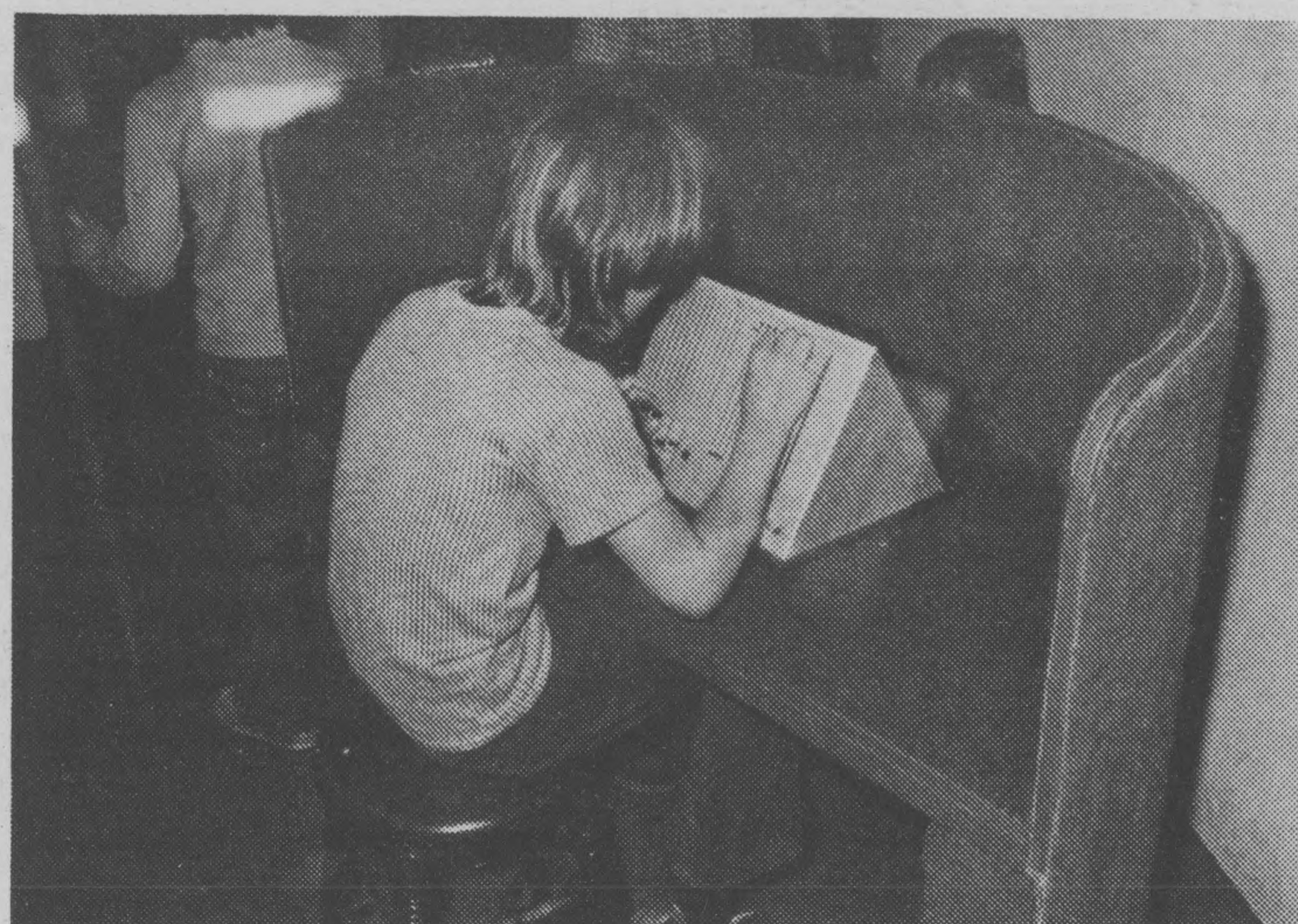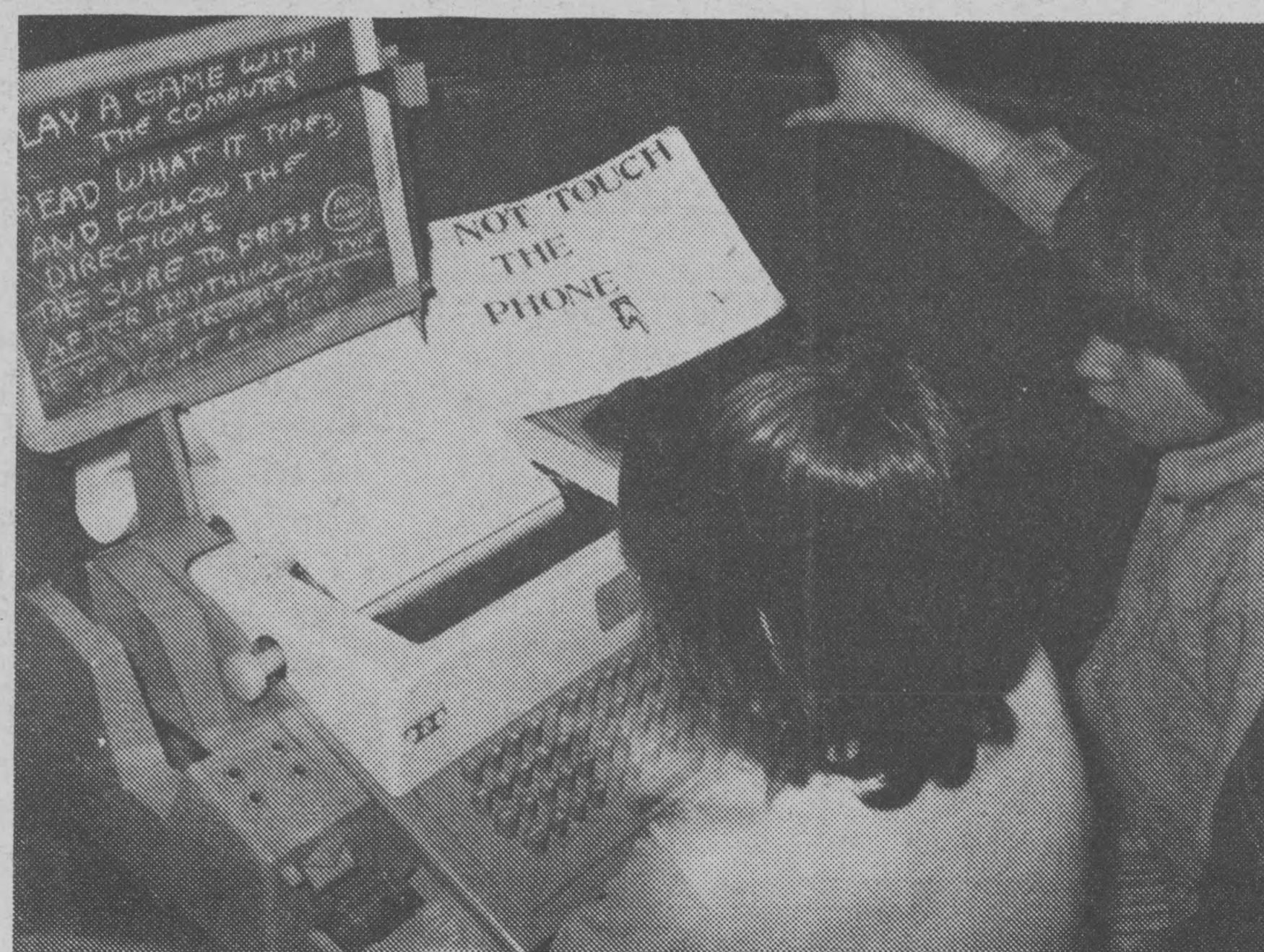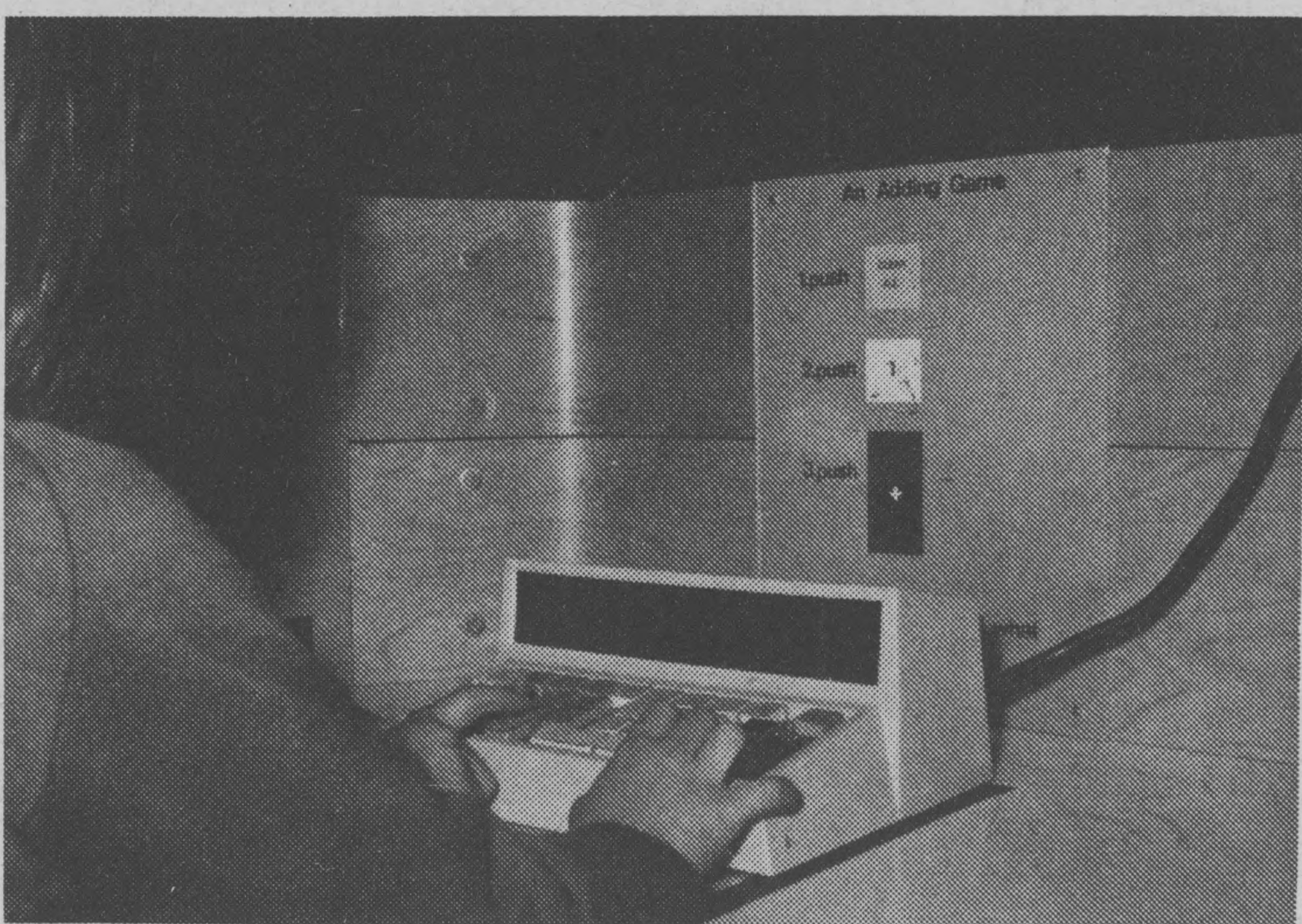

photos courtesy Childrens Museum

Though it sounds like just a lot of fun, the Children's Museum's involvement with computers is based on some really serious problems. Among them: 1] most people have very inaccurate conceptions of what computers are and what they can do; 2] many people (especially adults) feel an increasing level of dehumanization and fear as an outgrowth of expanded and often insensitive use of computers by government, business, industry, and the educational system; and 3] most of these feelings and conceptions are a direct result of the inaccessibility of computers for "average" people.

To date, over two hundred and fifty thousand people have visited the Museum's computer place. About 60% of them are children under the age of 15; the remainder are adults who often seem to hitch a ride when their children visit the Museum and who can therefore use the Museum as a comfortable, non-threatening source of information for their own needs.

The Children's Museum's Visitor Center is well-known in education circles for its development of exciting, self-directed "beginnings" — experiences designed to turn people on rather than teach them in the formal sense of that word, and its Computer Center is designed for the same sort of activity. "We do not teach programming," 21 year old Bill Mayhew, who withdrew from MIT a year or so ago and now coordinates the development and directions of the Computer Center, remarks. "There are other places that can do that better than we can."

The people who work in the exhibit space are responsible for its day-to-day operation and for observing what happens in the exhibit, for generating new ideas of things to try, and for helping each idea become a reality. They are college students from schools across the country with work-study programs, usually with no background in computer technology, who come to Boston for three month terms at the Museum. "Having people like that is particularly useful when it comes to things like making the equipment self-running and usable without the need for trained staff intervention and supervision, providing the visitors with the opportunity to use the machinery directly, all by themselves," Bill claims.

The Computer Center is open to the public during the Museum's normal hours, Tuesday through Friday from 2 to 5 PM, Saturday and Sunday, Boston Public School holidays and vacation days from 10 AM to 5 PM, and Friday nights for free from 6 to 9 PM. (Admission to the Museum at times other than Friday night is $1.50 for adults and $0.75 for children 3 through 15; there is no additional charge for use of the computer facility.) School groups from throughout New England come Tuesday through Friday mornings for special introductory programs, and special education groups come on Mondays.

Bill's hopes for the future of computers at the Museum hinge on obtaining foundation grants and donations. Right now, the facility costs $4500 per year to operate and maintain, and there is no money left for trained staff support or new equipment. Bill is very much anxious to explore ways of diverting money from equipment to people . . . "sort of the opposite of what most people seem to be doing nowadays."

A long-awaited and much-haggled-over proposal to the National Science Foundation was recently rejected. "That was a big blow to us, especially since we had been getting estimates of success ranging as high as 95% from people who knew NSF. We think the big reason for the rejection is the econopolitical situation in Washington. We hope to resubmit in June — meanwhile, we have a proposal in to the National Institute of Education, and we're writing a small pilot-project one for the Sloan Foundation. As far as we can tell, no one in North America is doing anything quite like what we're trying, although Seymour Papert's work at the LOGO Project at MIT, Resource One in San Francisco, and several other places are doing pieces of it."

Bill says that the Museum is always interested in helping other people get started, "and in getting help for our own work," and suggests that interested people contact him at the Children's Museum.

The Children's Museum
the Jamaica way
Boston 02130

NICE WORDS ABOUT US

REVIEWS

The People's Computer Company is a newspaper published 5 times during the school year. It is "about having fun with computers and learning how to use computers and how to buy a minicomputer for yourself or your school and books...and films...and tools of the future." The first issue, published in October, 1972, contains book and film reviews, a description of programs developed by the Huntington Project, computer games, a miniprimer on strings, ways to get the entire school involved in your computer education program, descriptions of hardware, an article about what's wrong with BASIC, ways to repair paper tape, cheap tape winders and writing music in BASIC. As you can see, it's well worth the subscription price of $4 (reduction on group orders) from People's Computer Company, P.O. Box 310, Menlo Park, Calif. 94025.

A bright orange paperback entitled My Computer Likes Me (when I speak in BASIC) differs from other manuals or teaching devices in that it is about "people, computers, and a programming language called BASIC." The book is permeated with an attitude about the relationship between people and machines, and it never loses sight of the humanity of the reader. It is full of diagrams and the print varies in size and format to hold the reader's attention. Some of the section titles are "Mistrakes," "Sorcerer's Apprentice," and "The Handy-Dandy Super-Versatile For-Next Loop." It introduces files, subscripts, and data bases. There is a brief list of "Books We Like" at the end.

The best features of this book are its exciting, readable format, its suitability for many systems (it mentions as it goes along how your particular system may be different) and its urging of the reader to discover for himself by trial-and-error and experimentation. It is altogether a refreshing experience in a world of stuffy, dull texts.

This book may be ordered from DYMAX, P.O. Box 310, Menso Park, California 94025 for $1.19 a copy.

THE "UNDERGROUND" LOOK OF STARS, CARTOONS AND "MOD" CALLIGRAPHY IS THE SHOWCASE for the computers-for-fun philosophy of the "People's Computer Company," a tabloid-size newspaper launched last fall by Dymax as a how-to-tool for computer use and minicomputer purchasing. Aimed primarily at teachers of eight-to-18-year-old students, PCC is edited by Bob Albrecht of Dymax on a five-times-yearly basis.

A major plank of the publication is its advocation of Basic for young programmers, a spin-off of the 1965 Dymax protest against oversophisticated programming for young students which culminated in the firm's organization of The Society To Help Abolish Fortran Teaching (Shaft). Game-type programs included in current issues include advice on what mini they would run on, and changes required for using a different machine. The publication is available for a $4 subscription from P.O. Box 310, Menlo Park, Calif. 94025.

DYMAX PUBLISHES BASIC GUIDEBOOK

My Computer Likes Me When I Speak in BASIC is a neat little book published by DYMAX of Menlo Park, Cal. Its purpose is to teach an absolute novice how to use a computer terminal and program in BASIC. It does this in an easy, carefree way and is designed for use as one sits at a terminal.

The book would probably be excellent for a person who has had no kind of experience with a terminal. It won't take him too far, but it will get him started and he will probably have fun! Copies can be obtained for $1.19 each from DYMAX, PO Box 310, Menlo Park, Cal. 94025.

PUNCHLINE

The Newsletter of LACE
La Crosse Area Computers in Education

University of Wisconsin
La Crosse, Wisconsin 54601

Saturday Review of Education

People's Computer Center. "Gotcha!" cries a twelve-year-old as he "shoots down" an enemy spaceship blipping across a computer screen. Another computer plays football and is on the short end of a 35–0 route at the hands of a dignified middle-aged man. My computer plays "guess a number." I guess three-digit combinations; it signals "bagels" if none of the digits are correct, "pico" if a digit is correct but in the wrong place, and "fermi" if a digit is correct and in the right place. I try five times, and it types, "You got it."

Bob Albrecht runs this workshop. It's much like his People's Computer Center in Menlo Park, which, he hopes, will be a prototype for setting up "friendly neighborhood computer centers" everywhere. "We'd like people to think of us as the local bowling alley —a place to come have fun," says Albrecht. What about the schools? Another workshop leader says nearly a million students in grade school and high school are now getting some part of their education from computers. "A computer can respond moment by moment to the fluctuations of a kid's curiosity," he says. "Kids can learn to program computers as early as fourth grade."

I had reopened doors into ways of learning that I had shut or that had been closed upon me for years.

Bead Game Music. We begin by constructing cardboard wheels with twelve spokes; on each spoke is a red, blue, yellow, or green bead. It's a musical abacus: push some beads out to the rim and you've got a chord. "This is a way of visualizing twelve-tone music so you know what it's going to sound like as you write it," says Peter Lynn Sessions, Zephyros member, a computer scientist and former rock musician. "With the bead game an absolute beginner can learn to compose twentieth-century music."

JANUARY 1973

Sunflower Source. Mike Young, looks as if he weighs 180 pounds, le onto a cardboard table and jumps and down. It holds firm. The demstration is Mike's way of advertis the strength of tri-wall—a three-ered, corrugated cardboard. Close a young woman is using an electric to cut out a table top in the shape a dolphin. Someone else carries a new easel. Mike runs a resource center for teachers ("Sunflower Source that carries tri-wall, toys, and ot supplies for schools." Using tri-Mike says, kids can make themsel what they use in their own classroo "Teachers have got to get the hell of the abstract and into the physical the kids can see." I decide to m three bookcases. That'll take two she of tri-wall at $2.55 a sheet.

Graffiti scrawled on the "I wish" w "I wish this was 1947 and Pen Sch was in Marshall, Illinois, and I going to it."

While I am still cutting up bo shelves, the fair comes to its close am reluctant to leave it, and I rea why: I have reopened doors into w of learning that I had shut or that been closed upon me years ago.

People's Computers *Learning, MARCH 1973*

Though touted as superservants designed to relieve us of mental drudgery, computers interact with most of us only as adamant bill collectors. Two organizations in California's San Mateo County, however, are working to make computers more accessible.

The phrase "Dymax Computer Center" conjures up visions of crew-cut men in white lab coats, banks of cathode ray tubes and all the other appurtenances of a futuristic educational technology. In reality, Dymax, or the "People's Computer Center," is more like an "our gang" clubhouse, circa 2001. It's a big, cluttered warehouse, full of posters, electronic circuitry, frayed program punch tapes—and kids.

For all the informalities, Dymax is seriously concerned with the educational uses of computers. Says Bob Albrect, director of the center: "Dymax was established because too many people are teaching computers to control kids. We want to teach kids to control computers." Drill and practice in a conventional computer-assisted-instruction sense is anathema to the people at Dymax. Instead, they concentrate on developing elementary programming skills in kids as young as 12. The children then use the Dymax terminals for problem solving and for playing games from warfare to economics.

Computer time at Dymax costs about two dollars an hour, less than a tenth of the cost of commercial computer outfits.

A growing number of adults are playing computer games at Dymax along with the kids, renting the terminals in much the same manner as you rent a tennis court or a pool table. Albrect feels that this kind of recreational use of computers will become a major leisure activity in the next ten years. He is working on plans for a coin-operated computer terminal that may show up in the not-too-distant future right alongside the village pinball machine.

PCC Uses Free-Style Tabloid Format To Jolt Student, Teacher DP Thinking

By Don Leavitt
Of the CW Staff

MENLO PARK, Calif. — In contrast to many slick-paper pedagogical journals, *People's Computer Company* is a free-wheeling, tabloid-size newspaper that could be mistaken for junk mail at first glance.

But the contrast is perhaps more apparent than real. Sprinkled liberally with stars, arrows, fanciful calligraphy and cartoon figures, PCC (vol. 1, no. 2), still has a substance that belies its "underground" appear-

Education

ance.

The paper was started last fall by Dymax, a teaching and software firm, to help readers learn how to use computers, and how to buy minicomputers. It also lists books, films and music and "tools of the future," all with some relationship to computers.

But more than anything else, "editor" Bob Albrecht ex-

plained, PCC is a newspaper about having fun with computers. They needn't be, and shouldn't be, objects of fear, especially to the eight- to 18-year-old students, and their teachers, to whom PCC is primarily addressed.

PCC and Dymax generally advocate the use of Basic for the youngsters and several game-type programs are in the current issue. The authors indicate what mini the programs run on and, in some cases, what changes would be required to put it on a different machine.

As long ago as 1965, Dymax organized a group to protest oversophisticated programming for schoolagers, and the Society to Help Abolish Fortran Teaching (Shaft) is still active. A major portion of the current issue is, however, devoted to a serious if somewhat rambling "polemic" explaining what author Marc LeBrun sees as flaws in Basic, both as an educational tool per se and as an example of "computer science."

Detailed Pricing

Continuing a presentation started in the first issue, on the Huntington Project funded by the National Science Foundation and conducted at State University of New York at Stony Brook, PCC provides a detailed price list of the simulation programs and related manuals, now being distributed by DEC.

The editors also provide a page of terminal terminology, distinguishing between hardwired, single-user "dial-up," and multiple-user "multiplexed" terminals. But they also illustrate a storage cabinet from Sears which they use for paper-tape storage, and they describe (tongue-in-cheek) how a movie projector reel can be used as a paper tape winder.

Mixed in with such patent nonsense, PCC notes that in the next few issues space will be given to discussions of how to write bid specifications for minicomputer systems for classroom use. The editors intend to describe some of their own experiences in this area, but they are seeking contributions from others.

They are interested in both hardware and software specifications, and expect to identify equipment or programs by name if that is deemed appropriate.

Dymax will publish PCC five times a year, and it is available on a $4 subscription, from P.O. Box 310, 94025.

FOR THE PEOPLE

A People's Computer Center, offering time for $2 an hour, opens this month in Menlo Park, Calif. Open to the public, it's dedicated solely to the recreational and educational use of computers, and will have an 8K DEC EduSystem 20 with two terminals. There'll also be a Tektronix programmable calculator, several four-function calculators, and various gear on temporary loan from time to time. Plans are to rent calculators. The neighborhood, near Stanford Univ., is loaded with coin-operated Space War games in bars, stores and at the San Jose airport. But the PCC, slanted initially to youngsters, will have games that teach math and are fun. There'll also be a 6-week course for kids on Games Computers Play. It's all being run by Dymax Inc., educational consulting outfit that soon begins publishing its own newspaper for teachers.

DATAMATION

PEOPLE'S COMPUTER COMPANY

A new concept of computing is spreading, namely, bringing real, live, interactive computing to the masses. Helping this revolutionary idea to spread, i.e., using computers for people and not against them, are Bob Albrecht and Leroy Finkel of the People's Computer Company.

PCC, an open computer workshop in Menlo Park, California, has kids (and adults) in it nearly 24 hours a day, 7 days a week "playing" games with a DIGITAL EduSystem 20. Number games, guessing games, simulations, moon landings, king of Sumeria, and all kinds of mind-expanding games are on tap at PCC. Or, best of all is when kids decide to write their own games—the enthusiasm has to be seen to be believed!!

PCC also puts out a contemporary newsletter called, appropriately enough, "People's Computer Company." Subscriptions are $4 (check or money order) for 5 issues to:

Bob Albrecht
DYMAX
P. O. Box 310
Menlo Park, California 94025

Be sure to mention EDU!

digital
edu

Digital Equipment Corp.
146 Main Street
Maynard, Mass. 01754

COMPUTER ACCESS

The Computer Freaks Are Coming. . . . There have always been high school and college students with an all absorbing interest in cars or electronics or model planes and rockets. Now a new breed of teenager is making a hesitant entrance on the stage - - the computer freak. Playing around with computers is almost a cult in certain university engineering schools and in some computer equipped professions.

With schools using computers more and even teaching basic computer courses the computer freaks are bound to emerge. An article in the December 7 1972 issue of *Rolling Stone* gives some practical tips on plugging into the educational computer scene.

Dymax, mentioned in the RS article, calls itself a "People's Computer Center" and includes among its services a newsletter "about having fun with computers, learning how to use computers, how to buy a minicomputer for yourself or your school, and books and films and tools of the future." They also have an informal workbook, **My Computer Likes Me**, that teaches BASIC -- an easy-to-learn, general purpose, education-oriented computer language.

MEDIA MIX
145 BRENTWOOD DR.
PALATINE, IL 60067

Media Mix is published monthly, October-May One-year subscription is $5, two years $9.

November 15, 1972

People's Computer Center opens

A People's Computer Center, offering computer time to the public for $2 per hour, is opening this month in Menlo Park, California. The system is dedicated solely to the instructional and recreational use of computers and will include: an 8K DEC EduSystem 20 computer with two terminals; a Tektronix programmable calculator, several four-function calculators; and other equipment. The Center will initially be slanted towards youngsters and will feature games that "teach math and are fun." Dymax, Inc., an educational consulting firm, will run the Center.

Microfilm equipment for loan

A similar project for community involvement with new technologies is underway at Evergreen State College in Olympia, Washington. Evergreen allows patrons to borrow videotapes, portapak video cameras, microfilm, microfilm readers, audio cassettes, movie films and projectors. And, next year they plan to circulate *computer terminals.*

In this issue, the last for this school year, we will "review" some music programs from other sources. There are two main themes to this article. First, to suggest the variety of approaches to "music" programming which are possible in BASIC. Second, to demonstrate a few ways to use *TABLES* of numbers to help solve music programming problems.

Here are the three program we'll be talking about in this issue. All come from different sources, and are "about" different aspects of music, and all use tables in some way.

PROGRAM NAME	SOURCE
KEYSIG	*Aardvark & Co., Lakewood, Colorado*
MUSIC	*Kemeny & Kurtz, 2nd Edition*
ANLSYS	*Linda J. Borry, TIES, St. Paul, Minn.*

KEYSIG is a program from Aardvark & Co., a writing team of high school students and other contributors dedicated to "programs to take up an absolute minimum of computer space and perform a maximum purpose." Among the many good programs contributed to HP's Public Library, the Aardvark team has produced several music programs. KEYSIG is representative of these products. For details on the rest, you should consult the HP BASIC Program Library Handbooks, or write directly to Aardvark:

Aardvark & Co.
2130 Bell Court
Lakewood, Colorado, 80215

Here's a RUN of KEYSIG —

```
RUN
KEYSIG

?CNFNGN
```

```
                !                !                !
----------------!----------------!-/--------------Ø-!----------------
                !                Ø !               Ø !
----------------!--------------Ø-!----------------Ø---!----------------
                  !              Ø !               Ø !
----------Ø-!-6---!--------Ø------!--------------Ø------!----------------
      Ø  !          Ø  !                Ø  !
----------Ø---!----------------Ø------!--------------Ø-------!----------
     Ø  !          Ø  !                          !
----------Ø----!----------------------!----------------Ø----!----------
     Ø  !          !                        !
   -Ø-

          CN              FN              GN
```

How does it work? The "brains" of the program is a table or array A. During the program RUN, this table is filled with numbers representing *characters* to be printed as the last step. The numbers refer to positions in a string, labeled C$.

```
    C$ = "CNGNDNANENBNFSCSFNBFEFAFDFGFCF - 0#6!!"
```

The first part of the string (characters 1 — 30) is used to identify the input key.

CDEFGAB = the seven letters of the traditional music "alphabet."

N = "natural", F = "flat", S = "sharp"

The last part is used for printing the resulting scale on the staff, along with the appropriate key signature.

Blank, —, and ! for the format of the musical staff.

Ø for the notes of the scale as they appear on the staff

(sharp) and 6 (flat) for the key signature.

The table A is used to control which of these characters in C$ is to be printed. The details of how this is accomplished would take up the rest of this article. Here's the main idea.

The variable A is assigned a value based upon the value of a "cell" in table A. Then the instruction

    PRINT C$(A,A)

does the rest. Naturally, this operation is done many times to produce the entire printout.

A complete listing of KEYSIG is published as HP Library Program A104-36258A (February, 1973).

MUSIC, by Kemeny & Kurtz, the designers of BASIC, is published in full with complete details in their book

BASIC PROGRAMMING
John Wiley & Sons, Inc.
First Edition, 1967

There is apparently no change in this program between the first and second editions of this book.

This program supplies a harmony in the key of C major for a melody which is given to the program as DATA.

The harmony consists of a chord progression which is then "arranged" by MUSIC according to the traditional rules of "voice leading." Kemeny and Kurtz list Piston's *Harmony* as the reference for the rules they used; the treatment of the harmonization is thus familiar in style. The program does quite well on "Swanee River," the example they present. It could be expected to perform less well on melodies whose implied harmony is more complicated. Here's the RUN:

```
RUN
MUSIKK
```

NOTE NUMBER	CHORD	S	A	T	B
28	1	Ø	-3	-5	-7
27	5	1	-1	-3	-1Ø
26	2	1	-2	-4	-6
26	5	1	-1	-3	-1Ø
25	6	Ø	-2	-5	-9
24	1	2	-3	-7	-14
23	5	4	-1	-6	-1Ø
22	6	7	2	-2	-9
21	6	5	2	Ø	-9
2Ø	1	7	2	-3	-14
19	4	Ø	-2	-4	-11
18	2	1	-2	-4	-13
17	6	2	-2	-7	-9
16	1	Ø	-3	-5	-7
15	5	1	-1	-3	-1Ø
14	1	2	Ø	-3	-14
13	5	1	-1	-3	-1Ø
12	6	Ø	-2	-5	-9
11	1	2	-3	-7	-14
1Ø	5	4	-1	-6	-1Ø
9	6	7	2	-2	-9
8	6	5	2	Ø	-9
7	1	7	2	-3	-14
6	4	Ø	-2	-4	-11
5	2	1	-2	-4	-13
4	6	2	-2	-7	-9
3	1	Ø	-3	-5	-7
2	5	1	-1	-3	-1Ø
1	1	2	Ø	-3	-14

These are the notes. Only natural notes are used — so 0, 1, 2, 3, 4, 5, and 6 stand for the notes C, D, E, F, G, A, and B.

    0 = middle C
    1 = D above middle C
    —1 = B below middle C

MUSIC uses two tables, one to control the chord progression and one to ensure that voice leading is "correct" according to the rules.

First, the chord table, P. Here's how it looks after it has been filled from DATA lines.

chord	1	2	3	4	5	6	← order of preference
1 ↓	5	4	1	6	2	3	
2	6	4	2	1	5	3	
3	6	4	3	1	2	5	
4	1	5	4	6	3	2	
5	2	6	5	4	1	3	
6	1	5	3	6	2	4	

How the table works:

The program performs the harmonization by working from the last note backwards to the first note. It uses the P-table to govern its choice of the next chord to *precede* the current chord. Six different chords are used —

1 = Cma   2 = Dmi   3 = Emi   4 = Fma   5 = Gma   6 = Ami

The *rows* of the table above are labeled with the number of each chord. The program uses the row representing the current chord to "look up" the preferences for the next. The columns of the P-table represent the order of preference (from traditional theory) for the chord to precede the given chord. For example, look at row 2 of the table —

2   6   4   2   1   5   3

This means that chord number 6 (Ami) is the first choice to precede a 2-chord (Dmi); the second choice is a 4-chord (Fma), the third is another 2-chord, etc.

The result is that this table has stored a *theory* of chord progressions. According to this theory, some chords are better than others at certain points in a progression. Note that the theory does not *exclude* any chord progression completely (within the six chords used), so that any progression is possible. The higher-numbered preferences are merely *unlikely* to occur.

The program examines each prospective chord in order of preference in the table, and selects the chord with the highest preference (1 is first preference) that also contains the melody note at that point. The melody note is assigned to the soprano (highest) voice, and the root of the chord to the bass (lowest). Then, various possibilities for assigning notes of the chord to the alto and tenor voices are considered, and stored in another table, W. At this point, such considerations as "parallel fifths" are used to decide whether or not a given assignment is "correct" according to the rules of traditional arrangement. If no "good" transitions can be found, the program "backs up" and selects another chord further along the same row of the P-table used to select the rejected chord. If this fails, the program backs up another chord and tries again. In the RUN preceding, there was one case of backing up, at note 26 of the melody.

This program is a good example of how to translate a theory into a BASIC program. If you want the details, it's all in the book.

Next, a table program which can be used to analyze and (potentially) control *melodic* progressions, or movements from tone to tone of a single line of tones. This program is called ANLSYS, and was written by Linda J. Borry at TIES. For further information on this and other good music programs, contact PCC (that's us) or write —

Linda J. Borry
TIES - Minnesota School Board
1925 W. County Road, B2
St. Paul, Minnesota 55113

(Try to be sensible about requests. We're all busy, Linda more than most, we suspect.)

ANLSYS is a "simple" program, but extremely useful to music programmers. You input the notes of a melody (any sequence of tones), and the program computes the *percent* of times each note was followed by each other note. The result is stored in a 12 x 12 table, and the table is printed. Here's a sample RUN —

```
OPEN-ANMAT,2
RUN←←RUN
ANLSYS
?C
?D
?C
?C
?D
?E
?B
?C
?C
?E
?G
?D
?F
?C
?A
?B
?D
?G
?D
?G
?END
```

	D	D#	E	F	F#	G	G#	A	A#	B	C	C#	
D	0.0	0.0	16.7	16.7	0.0	33.3	0.0	0.0	0.0	0.0	33.3	0.0	6
D#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
E	0.0	0.0	0.0	0.0	0.0	50.0	0.0	0.0	0.0	50.0	0.0	0.0	2
F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	1
F#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
G	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
G#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
A	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	1
A#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
B	50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.0	0.0	2
C	33.3	0.0	16.7	0.0	0.0	0.0	0.0	16.7	0.0	0.0	33.3	0.0	6
C#	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0

The rows and columns of the table are the twelve tones of the chromatic scale. This program is thus useful for many styles of music, tonal and atonal.

In our example, a random melody in the key of C major, tone D (first row of the table) was followed by tone E 16.7% of the time. Tone F (fourth row of the table) was followed by G 100% of the time. See how that works? The rightmost column of the table gives the *total* number of times the row note appeared in the melody.

That ain't all! The program stores its cumulative results in a file (named ANMAT, 2 records). You can thus use the program at several sittings to build a large volume of data about a particular melodic style. Linda has used this program to analyze the melodic progressions in Bach chorales, reading in the melodic progressions from "about 30 chorales." Another of her programs, BACH, uses the cumulative table to write fair examples of "new" Bach chorales in four voices.

This approach lets you generate a theory of melodic progression "empirically," without prior rules. It could (we think) be used to analyze the melodic style of any composer who used the 12-tone system as the basis for composing (this includes most traditional and many modern composers). BARTOK, anyone?

Naturally, tone-to-tone movement is only one aspect of "melodic style," but it's a good start. Could such a table be constructed to analyze chord progressions? How about rhythmic patterns?

We hope you have enjoyed our BASIC MUSIC articles this year. Next year, more 12-tone theory and, hopefully, bigger and better composing programs in BASIC. So long for now.

BUILDING

BUILDING A PUBLIC LIBRARY

If you've been following our past ramblings you have seen us repeatedly say that computers should be for all the people in your community (school), not just a select few. In most school situations we have witnessed what I consider to be a *maturation* period in computer usage:

Infancy — The computer is hoarded by computer-types exclusivley either hiding in the math department or some data processing corner.

Early Childhood — One department (usually math) requires all students to learn "some" programming.

Puberty — Early childhood is reappraised and another department starts teaching programming and lets students solve homework problems on the computer (usually science).

Adolescence — Science and math departments let students use the computer for something other than programming, such as simulation activities, or using "canned" programs written by others. These new users do not necessarily know how to program.

Maturity — Social studies and business education departments and the rest of the school now use the computer as a utility . . . to solve problems where appropriate; to teach programming where important; classroom simulations; statistical analysis; using public library programs of all varieties including games.

The time line for these five stages may be anywhere from one to five years (or never). It largely depends on how quickly YOU want maturity to occur . . . if you do at all.

It is my opinion that the best way to hasten maturity is to make it *easy* for people to get to your system and have plenty of programs that are easy for novices to use. The first step is simple . . . get your computer or terminal *out of the classroom* into a lab, resource center, library, or wherever it will be readily available all day long. If you place it in a math classroom, business teachers will be reluctant to cross the threshold of that room (and vice-versa). Can you imagine a social studies teacher in your physics lab? Don't force people to cross these "invisible" barriers. Make the computer *available*.

Step 2 . . . *plenty of programs* is the subject of this article. Today, there is a massive quantity of programs available in BASIC geared to educational use. You need them on your disk or available on paper tape for your teachers and students to use. We'll call these programs your PUBLIC LIBRARY.

Finding these programs is just one of the problems. *Evaluating* their applicability, *modifying* them to fit on your system, *loading* them in (make a spare tape) and *preparing documentation* so others can use the program is a real time consuming operation (it is a rare day when I can find a program that will go directly into our system without some help). A school district could assign someone full time to this task and the job still wouldn't get done. As it is, the job isn't assigned to anyone full time and it is left to chance (the vendor) or the deliriums of the computer teachers to build the library on their own time. Well gang, here's my contribution to make your job easier. I hope others of you will contribute ideas to make mine easier as well.

16

PUBLIC LIBRARY

Here is where to find programs.

HEWLETT-PACKARD
11000 Wolfe Road
Cupertino, CA. 95014

HP provides a library of 200 programs or so with each educational installation. These include the original Huntington Project programs plus an assortment of "contributed" programs. (Quite frankly, I scrapped about half of this assortment as not applicable for secondary schools or too complicated to use by novices.) Customers also get a listing of the "regular" HP BASIC library plus a new set of programs from Stanford Business School. Many of these programs will be usable in your school. Many more will not (the Stanford programs are good but useless for secondary school). Notice you get *listings*. Paper tapes will run you $10 each. Periodically you will get updates and additions to the library from HP. How do *you* get these programs? Join the HP Educational Users Group as a full member ($15/year), and you will receive all their library listings and updates free . . . as part of your membership.

DEC
146 Main Street
Maynard, Mass. 01754

DEC's assortment of materials changes so often we're reluctant to list them here. There is a booklet entitled *Educational Program Index* which describes some 40 programs in FOCAL and FORTRAN and over 100 programs in BASIC. These include the original Huntington programs plus a variety of other small programs contributed by their educational users. Listings of all of these programs are available to DECUS members (DEC's users group) for free ($1 per paper tape). *You* can get listings of the programs by buying DEC's *BASIC Application Program* booklets that are available at very reasonable prices. The best way to keep current on DEC's materials is to subscribe to EDU (free), their periodic magazine. DEC plugs all its' new materials into EDU.

REMEMBER — DEC is the exclusive source for Huntington Two program tapes and materials.

As far as I can tell, DEC does not offer the grand variety of general purpose BASIC programs that HP has but they do have a good assortment of educational programs. (DECUS has a variety of programs available but they are mostly not in BASIC.)

TECNICA
3301 Vincent Dr.
Pleasant Hill, CA 94523

Yes, they're back . . . as a mail order instructional materials source! Tecnica made a deal with Northwest Regional Education Labs in Portland, Oregon to reproduce their REACT computer materials. While we really don't like much of the REACT materials, there is a book chock full of programs for your library. It's Course 2, Book 6, "REACT Computer Program Listings" for $2.95. It's a book of listings only — documentation comes in six books called "Application Units" which run $1.95 each ($2.95 for English). I wouldn't bother with the English book — it describes one huge poetry writing program which really turned me off.

CONTROL DATA
c/o Ms. Leah Krug
Rm. 2005 HQRC2E
P.O. Box 0
Minneapolis, MN. 55435

Who? Control Data! What do they know about educational computers? Not much, we suspect, but back in the archives of CDC is a man by the name of Dr. R.E. Smith (Doc Smith) who has a most phenomenal imagination and a prolific pen. As a result, CDC sells a neat series of booklets, *The Computer Explorer Series*, that serve many purposes. They contain listings of a variety of interesting and unusual computer programs (that's why we mention them here!). The same booklets teach advanced programming techniques using very original ideas. It takes a pretty sharp student to keep up with Doc Smith's explanations but the booklets are worth your reading as they may give *you* some ideas. Look at these titles:

Biorhythm Theory	Distances — Here to There
Interest in Money	Game Theory
Primes and Factors	Statistical Procedures
Factorials	Your Handwriting
Random Numbers	Big Arithmetic
Population Holocaust	Sorting Data
Excursion in Astrology	Metrology
A Look at Numerology	Visual Elements
Linear Programming	Simulated Gaming

They are $3 each with discounts offered if you buy the whole set. Doc Smith's prolific hand has written many other things including some beginners BASIC books. The latest, *A Visual Approach to BASIC,* is one worth avoiding. The ideas are "cute" (many borrowed from the Explorer Series . . . or vice versa) but poorly put together — I have grave doubts that any beginner is going to learn BASIC from it. Save the $5 for that book and spend it on future Explorer Series books that are sure to appear.

Ontario Library Review

PEOPLE'S COMPUTER COMPANY
P.O. Box 310
Menlo Park, CA. 94025

Subscribing to PCC will provide you with a listing of our game programs, some from Lawrence Hall of Science and other games contributed by our readers. And we'll try to keep you abreast of what's new in programs and where you can find them.

Where else can you get programs? We've started a mutual program rip-off (exchange) with two neighboring districts. Palo Alto and San Mateo High School Districts. You should do the same with your neighbors! Once we put our programs on the system, we prepared an INDEX program so that all users would know what's available. The list below is our GAMES section.

Are there more sources of program? Write us and let us know if you know of any.

LeRoy

```
GET-$INDEX
RUN
INDEX


INDICATE THE NUMBER OF THE PROGRAM LIST YOU WANT
1. BUSINESS ED
2. GAMES
3. MATHEMATICS
4. SCIENCE
5. SOCIAL SCIENCE
6. STATISTICS
7. UNCLASSIFIED
8. ALL OF THEM

CHOICE?2


APR  18   1973
```

Join the information conspiracy!!! P.C.C.

Write P.C.C. a letter or something

```
GAMES

ABAGEL-LETTER GUESSING GAME
AMAZIN-PRINTS A MAZE FOR YOU TO SOLVE
BAGELS-NUMBER GUESSING GAME
BATNUM-BATTLE OF THE NUMBERS GAME
BIOSIN-CHARTS YOUR LIFE CYCLE
BIOSUM-BIOSIN FOR TWO PEOPLE'S COMPATIBILITY
BLJACK- MULTI PLAYER GAME OF BLACKJACK
CAL- MAKES A 12 MONTH CALENDER FOR ANY YEAR
CIVIL-FIGHT 14 RECREATED CIVIL WAR BATTLES
GOLF- SIMULATED 18 HOLES OF GOLF
HAMUR-ECONOMIC SIMULATION OF A SIMPLE SOCIETY
HANG-THE GAME OF HANGMAN
HURKLE-FIND AN OBJECT IN A SQUARE GRID
KING-SIMULATION OF BASIC ECONOMIC SOCIETY
KNIGHT-MOVE A CHESS KNIGHT AROUND A CHESS BOARD
LETTER-PRINTS LARGE LETTER MESSAGES
LIFE-CONWAY'S GAME OF CELLULAR GENERATION GROWTH
LUNAR-TRY TO SOFT LAND A SPACE SHIP ON THE MOON
MARKET-SIMULATES THE OPERATION OF A COMPANY
MATCHS-GAME OF 23 MATCHES
MUGCIR-FIND AN OBJECT IN A CIRCLE GRID
MUGWMP-FIND MULTIPLE OBJECTS IN SQUARE GRID
NUL-NUMBER NEUTRALIZING GAME
NUMBER-NUMBER GUESSING GAME
ONEMUG-FIND AN OBJECT IN A SQUARE GRID
QUBIC- THREE DIMENSIONAL TIC-TAC-TOE
QUBIC5-5 LEVEL DIFFICULTY QUBIC
SIMNIM-SIMPLE GAME OF NIM
SLOT-SLOT MACHINE SIMULATION
STARS- NUMBER GUESSING GAME
STREK-THE FANTASTIC GAME OF STARTREK
STOCK-STOCK MARKET SIMULATOR
STOCKS-STOCK MARKET SIMULATOR (WITH PATTERN)
SUNSGN-PATTERNS YOUR PERSONALITY
TRAP-NUMBER TRAPPING GAME
ZEROES-LOGIC GAME

DONE
```

EduSystem

In case you missed the last two issues . . . EduSystem? An EduSystem is an educational computer system made and sold by DEC, Digital Equipment Corp., the largest mini computer manufacturer. EduSystems are built around the PDP 8 and PDP 11 mini computers. In looking over DECs catalog, I see lots of EduSystems — like Edu 5 and Edu 10 and Edu 20 and so on. Some use the PDP 8 computer and some use the PDP 11. This time I'll talk about some, but not all of the PDP 8 EduSystems. They let you start small and then grow bit by bit as your needs grow. Like this →

In the last two issues, I talked about Edu 10 and Edu 20. Now I'll say a few things about Edu 25.

Edu 25 is a self-contained time sharing system for up to 8 users (using BASIC).

★ Edu 25 BASIC has *string* variables and functions.

★ Edu 25 lets you store programs *on-line*. When you want a program stored on-line, you simply ask for it by name and !ZAP! it is loaded into memory ready to use.

★ Edu 25 lets you create and use *data files*. Data files may be used by more than one program.

Edu 25 consists of Edu 20 with 16K* words of memory plus two *DECtapes*.
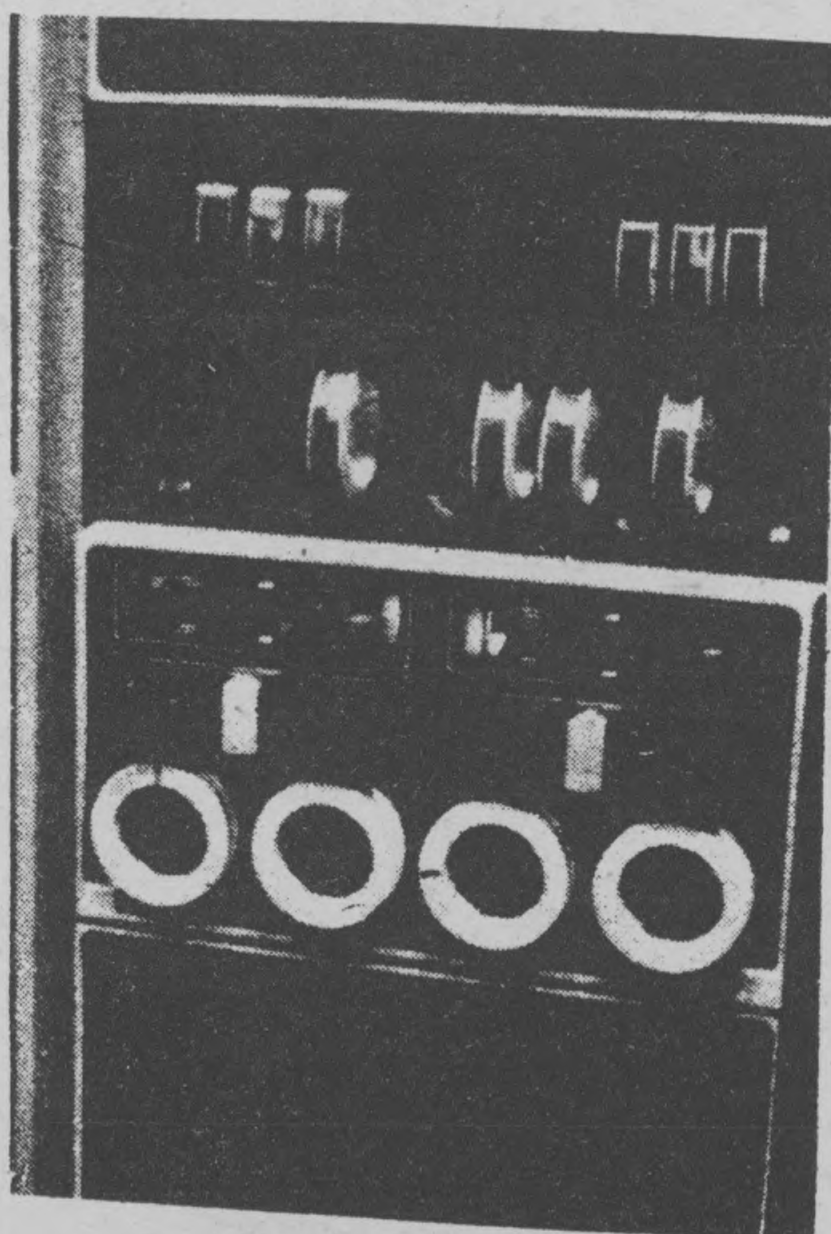
DECtapes are magnetic tapes on 5" reels. Each DECtape can store up to 188,672 words.

This is a dual drive DECtape unit. Two DECtapes can be mounted at one time. In this picture, you see four reels because for each DECtape feed reel there is also a take-up reel.

* Well — there is also a smaller Edu 25 that permits up to four users. It consists of Edu 20 with 12K + DECtapes.

PDP-8 BASED EDU-SYSTEMS

50 and 55 — 40 and 45 — 25 — 15/B & 30 — 20 and 21 — 15 & 30/I — 5 and 10
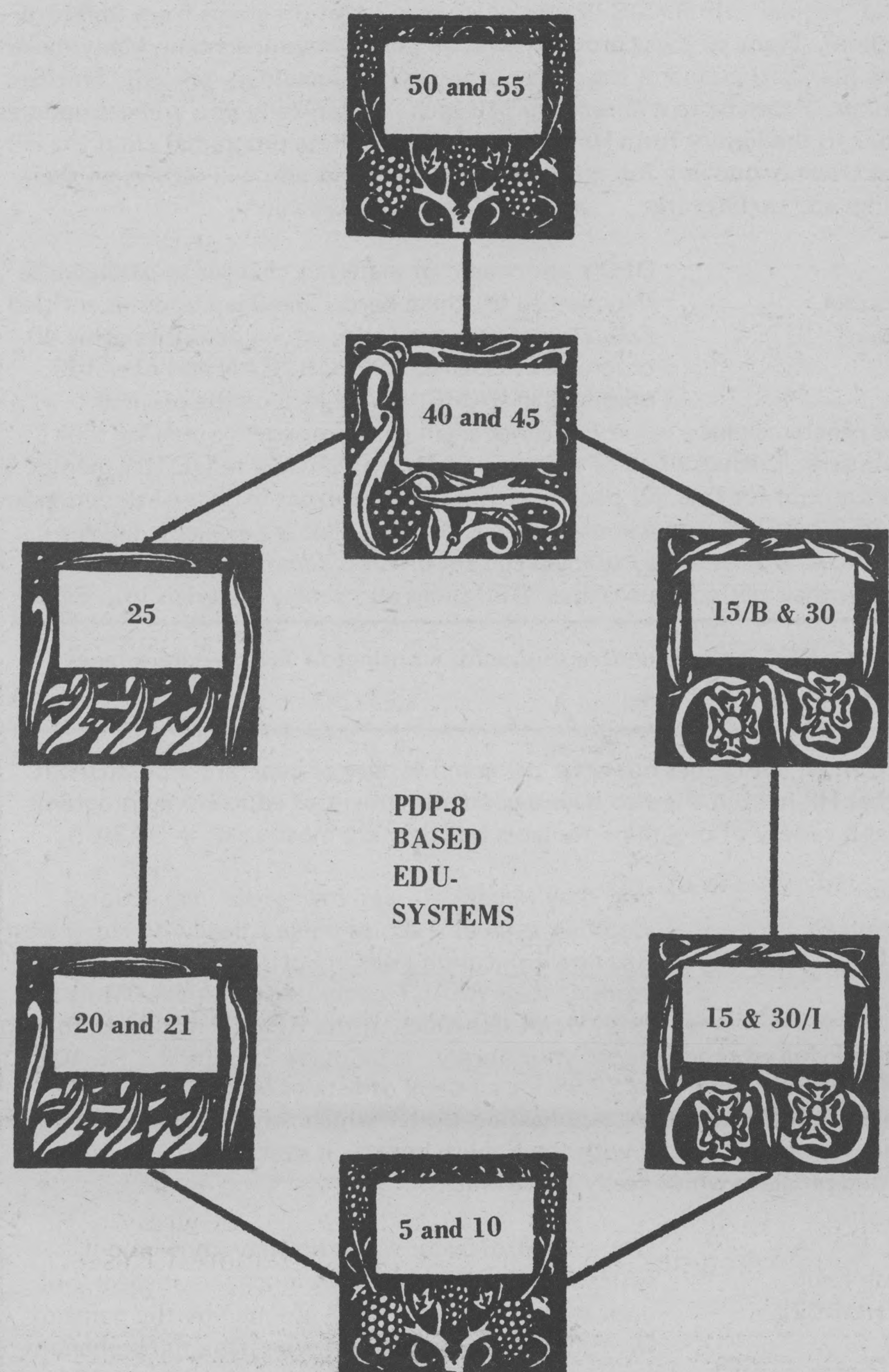
When you buy an Edu 25 you get all these pieces!

DEC MYSTERY NUMBER	NAME OF PIECE	PRICE	INSTALL CHARGE*	MONTHLY MAINTENANCE
PDP–8E–AE	COMPUTER WITH 8K MEMORY	$5650	–	$80
MM8–EJ	ADDITIONAL 8K MEMORY	3900	$175	40
KP8–E	POWER FAIL DETECT & RESTART	250	60	2
MI8–EF	HARDWARE BOOTSTRAP	500	60	5
H960–BC	CABINET	650	–	–
KA8–E	EXTERNAL BUS	250	60	3
KD8–E	DATA BREAK	500	60	3
BE8–A	OMNIBUS EXPANDER	500	60	–
TCO8–HA	DECTAPE CONTROLLER	4500	240	20
TU56	DECTAPE DUAL DRIVE	4700	60	30
LT33–DC	TELETYPE	1620	120	30
QRE25–SC	EDU25 + OS/8 SOFTWARE	550	–	–
QRELP–01	TEXTBOOK KIT	100	–	–
	TOTAL	$23670		$213

* The INSTALL CHARGE applies only if you add equipment to an already existing system. So, if you are expanding from Edu 20 to Edu 25, don't forget to add the INSTALL CHARGE.

But DEC quotes a ★★★SPECIAL PACKAGE PRICE★★★ and also a monthly lease price —

Price	$23190
Monthly lease	512
Monthly maintenance	213

Remember — prices so far include only one TTY. You can add TTYs or other terminals up to a total of eight. Here are DEC's prices

DEC MYSTERY NUMBER	TERMINALS & INTERFACES	PURCHASE PRICE	MONTHLY LEASE	INSTALL CHG	MAINTENANCE
LT33–DC	TELETYPE	$1620	$36	$120	$30
KL8–E	TELETYPE INTERFACE	300	7	65	5
LA30–PA	DECWRITER 30 CPS TERMINAL	2795	62	120	30
LC8–E	DECWRITER INTERFACE	300	7	60	6
VT05–B	DECTERMINAL DISPLAY	2795	62	80	30
H308	NULL MODEM ADAPTER (REQ.)	60	2	10	–
KL8–EC	DECTERMINAL INTERFACE	350	8	60	5
KL8–EA	REMOTE MODEM INTERFACE	350	8	60	5
DF01–A	ACOUSTIC COUPLER (for use with LT33 or VT05)	425	10	–	8

The REMOTE MODEM INTERFACE lets you attach a BELL 103A Modem from Bell Telephone Company (or the equivalent) so that you can use the system remotely via telephone lines. (You also need a terminal at *your* end of the phone line, of course!)

OK — let's see how much for a system with 8 TTYs, all local (hardwired).

Edu 25 with one TTY		$23190
plus seven TTYs	7 x 1620	11340
plus seven interfaces	7 x 300	2100
		$36630

Three things for you to do —

(1) Compute lease prices (per month) for the 8 TTY system.
(2) Compute maintenance price (per month) for 8 TTY system.
(3) Consult earlier issues of PCC to find out how to save money if buying Teletypes.

Now we will do one more — an Edu 25 with 1 TTY, 1 DECwriter and 1 DEC terminal on-site (local, hardwired) plus 5 remote TTYs — all obtained from DEC.

Edu 25 with one (local) TTY	$23190
DECwriter & interface	3095
DEC terminal & null modem & interface	3205
Five remote modem interfaces	1750
Five LT33–DC Teletypes	8100
Five DF01–A acoustic couplers	2125
	$41465

And — you also need 5 Bell 103A modems (or equivalent). Contact your local Ma Bell Data Communications salesman to find out how much these will cost you.

Again, consult earlier issues of PCC to find out how to save money on TTYs and acoustic couplers.

Well, that's as far as I am going with EduSystems. You can carry on — get the following book from DEC.

Edu System Handbook   ($5.00)

from: Digital Equipment Corporation
Communications Services, Parker Street
Maynard, MA 01754

And, if you can, get the following price lists from your friendly local DEC salesman.

* PDP8E, PDP8F, PDP8M END USER PRICE LIST (Jan, 1973)
* EDUCATION PRICE LIST (Jan, 1973)

Crunched BASIC for EduSystems

If you have an EduSystem, you can write your programs in good old standard BASIC or you can use what we call "crunched" BASIC. Crunched BASIC programs use up less memory space in the computer and can be read in via TTY paper tape reader in less time. Here are some features of crunched BASIC.

* More than one statement per line. Statements are separated by the reverse slash. For example, here is a complete FOR-NEXT loop on one line.

440 FOR S=1 TO 7 \PRINT "*"; \NEXT S

* Strings in INPUT statements.

140 INPUT "WANT THE RULES (1=YES 0=NO)";Z

* Statement following "THEN" in an IF-THEN statement.

290 IF G<X THEN PRINT "YOUR GUESS IS TOO SMALL"

Well, here is a complete program using some of the features mentioned above along with a couple of programming tricks to produce a very compact version of STARS (see PCC, Dec., 1972, page 3).

```
100 REM *** STARS - PEOPLE'S COMPUTER COMPANY
110 RANDOM
120 REM *** M IS NUMBER OF GUESSES ALLOWED
130 M=8
140 INPUT "WANT THE RULES (1=YES 0=NO)";Z\IF Z=0 THEN 310

200 REM *** HERE ARE THE RULES
210 PRINT "I AM THINKING OF A NUMBER FROM 1 TO 100. TRY TO GUESS"
220 PRINT "MY NUMBER. AFTER YOU GUESS, I WILL TYPE ONE OR MORE"
230 PRINT "STARS (*). THE MORE STARS I TYPE, THE CLOSER YOU ARE"
240 PRINT "TO MY NUMBER. ONE STAR (*) MEANS FAR AWAY. SEVEN STARS"
250 PRINT "(*******) MEANS REALLY CLOSE! YOU GET";M;"GUESSES."

300 REM *** COMPUTER 'THINKS' OF A NUMBER
310 PRINT\X=INT(100*RND(0))+1
320 PRINT "OK, I AM THINKING OF A NUMBER. START GUESSING."

400 REM *** GUESSING BEGINS. PLAYER GETS";M;"GUESSES
410 FOR K=1 TO M
420 PRINT\INPUT "YOUR GUESS";G\IF G=X THEN 610\D=ABS(X-G)
430 FOR J=6 TO 0 STEP -1\IF D>=2↑J THEN 440\NEXT J
440 FOR S=1 TO 7-J\PRINT "* ";\NEXT S
450 PRINT\NEXT K

500 REM *** DID NOT GUESS NUMBER IN M GUESSES
510 PRINT\PRINT "SORRY, THAT'S";M;"GUESSES. NUMBER WAS";X
520 PRINT "LET'S PLAY AGAIN."\GO TO 310

600 REM *** WE HAVE A WINNER
610 FOR N=1 TO 50\PRINT "*";\NEXT N
620 PRINT "!!!"
630 PRINT "YOU GOT IT IN";K;"GUESSES. LET'S PLAY AGAIN."\GO TO 310
999 END
```

For more info about EduCrunch BASIC, consult your EduSystem Handbook.

DATAMATION is a trade magazine (monthly) sent free to "qualified" computer users. Each issue is mostly articles written by professional computer people for other professional computer people — plus news of the computer industry, new hardware, new software, people, places, things, stuff.

Once in a while they print something of interest to a wider audience — here are two samples.

A philosophical essay
on the plausibility
of dividing by zero

The Fallacy
in
the Fallacy

by Jeryl W. Lafon

Mr. Lafon is a management analyst for the Bureau of Indian Affairs. He was previously ADP Coordinator for the Albuquerque district of the Corps of Engineers. He has had 10 years of experience in data processing and now specializes in dp standards and procedures.

It happens to virtually all computer programmers. If it has not yet happened to you, the odds dictate that someday it will. Your program has been running smoothly for weeks, months, maybe even years. But suddenly one day, out pops a tell-tale message on the console typewriter: ZERO DIVISOR!

Have you ever noticed, when the console type-out results from a limitation in the hardware or software, the message comes out in the form of some obscure code, like "ERR NO. X-15-P2A"? (When you look up the meaning of Error No. X-15-P2A, it probably says something like "UNDERFLOW EXCEEDS FIXED-POINT BOUND IN MIDDLE CURTATE OF REGISTER Q," which is a deliberately ambiguous way of admitting that Register Q is not adequate to handle numbers beyond the range of ±999. But when the message results from some microscopic oversight on the part of the programmer, the message comes out in plain English—DIVISION BY ZERO—so that all of the computer operators will know you goofed. ("Old Berkenheimer has been dividing by zero again—har, har, har!")

If you are lucky, the software has been designed to ignore the zero divisor and give you a correct answer in spite of the error message. If you are not lucky, you will have to modify your program to test for a zero divisor, and branch around the offending instruction in case the zero divisor occurs.

My question is, how much longer do we intend to put up with this indignity? Since it might be asking a bit much of the compiler writers, software experts, and hardware designers to handle such problems automatically, I am formally prepared to advance a plucky new hypothesis that could do away with this zero-divisor nonsense forever.

The mathematicians assure us that division by zero is not permitted. They mathematicians say (and not without cause) that when we divide by zero, mysterious things happen. Examples they are fond of using to illustrate the problem frequently start out by asking the student to assume that x = y. From this, a series of apparently innocent equations are developed (e.g., $x^2 = xy$, or $x^3 = y^3$), which end up by offering seemingly incontrovertible proof that 2 = 1 (or similar anomaly). After the student has been given time to ponder the enigma, the mathematicians will explain that the fallacy lies in some step where both members of the equation were surreptitiously divided by x — y, which must be zero since x and y are equal.

But it seems to me that there is a fallacy in the fallacy. (One of my computer associates, Mr. L. Eisenzimmer, refers to this as a *nested* fallacy.) Granted that the mathematicians have been operating under this premise for quite a while now, I nevertheless resent being told that I am not *permitted* to divide by zero, on such tenuous grounds as displayed in the usual examples. Therefore, I will attempt to show that dividing by zero is logical, desirable, and practical.

In order to clarify the ensuing discussion, I am introducing a new word to describe such concepts as *zero* and *nothing*. The word is *pragmadox*. A pragmadox (*pragmatic* + *paradox*) is defined as a concept which is inherently meaningless or self-contradictory, but which nevertheless has practical application. Imaginary numbers (e.g., the square root of −1) and geometrical points are classical examples of a pragmadox. Geometrical points, for instance, have been defined as locations in space, without size or shape. Now if a point has no dimensions, it cannot

be properly said to exist, except as a figment of the imagination. But without such concepts we could never have reached the moon.

The word *nothing* is another example of a pragmadox. *Nothing* can only be defined with respect to *something*—i. e., as the opposite of something, the absence of something, or as that which does not exist. Yet even *nothing* must seemingly be *something*, in order to be an "opposite" or a "that which." As the poet Wallace Stevens once put it, "There is not nothing; no, no, never nothing." Well, maybe there isn't and maybe there is. But in either case, we definitely need the concept.

Okay. If we select a quantity, x, and refuse to divide it by anything, the quantity (I think any competent mathematician will agree) remains the same. In slightly more mathematical terms,

$$x \text{ (not divided by anything)} = x$$
—or—
$$x \text{ (divided by nothing)} = x$$
—or—
$$\frac{x}{\quad} = x$$

Now if *zero* may be considered to be the mathematical equivalent of *nothing*, I see no great harm in expressing the foregoing bit of philosophy in an equation of the form, x/0 = x. But here, some mathematician will pounce gleefully on the fact that x/1 (is also) = x, and will accuse me of saying, in effect, that zero equals 1. (These mathematicians really know how to hurt a guy.) Well, in a peculiar sort of way, that may be exactly what I *am* saying—as I will attempt to demonstrate in a moment.
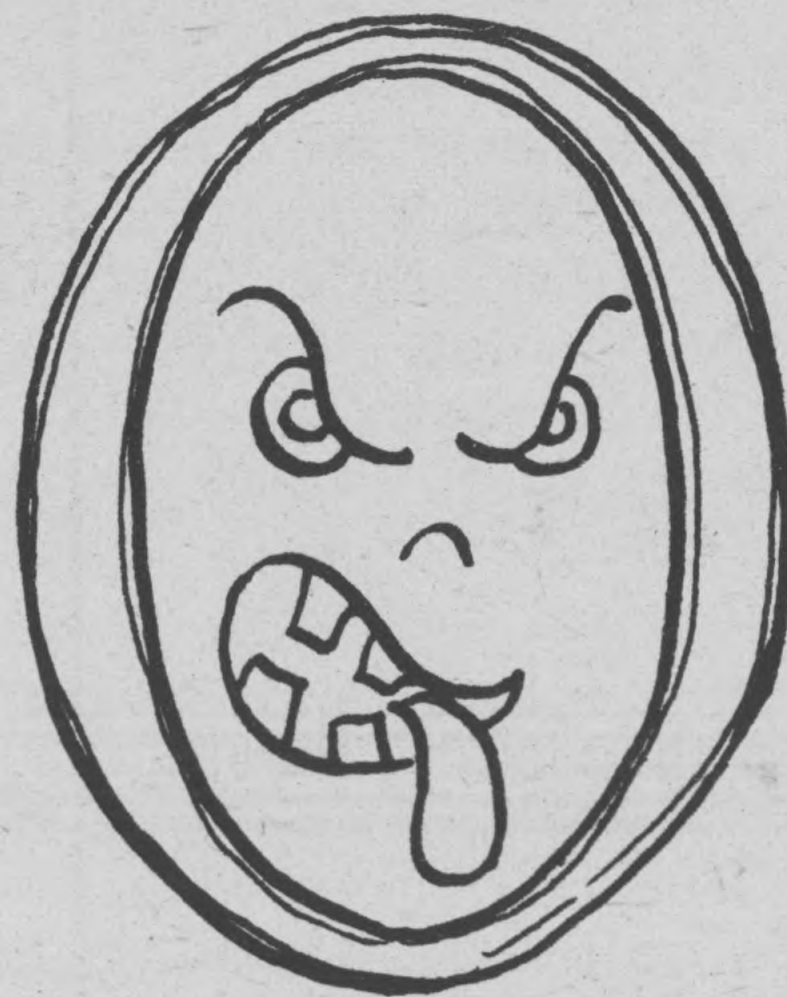
First, however, let's examine the problem from a slightly different angle. When we divide a number, x, into another number, y, we are trying to determine the minimum number of x's contained in y, without exceeding the value of y. If the numerator, y, happens to be zero, and the denominator happens to be 2, we are asking how many pairs of units are needed to equal zero. And the answer, of course, is *no* pairs. We need a total absence of units, and the answer is therefore zero. Conversely, when we divide a positive number by zero, we are asking how many non-units are needed in order to equal or closely approximate a given number of units. But it seems intuitively obvious that if Dick had no apples, Jane had no apples, and Spot had no apples—in fact, if everybody in the universe had no apples and pooled them all together—there still wouldn't be enough non-apples to make even one tiny fraction of a real apple. The answer, therefore, is *no* amount of non-apples (or non-units or zeroes). Thus the true quotient of x/0 is zero.

This answer may come as a shock to some readers. Being conditioned to the idea that the smaller the divisor the larger the quotient, such readers might suppose the quotient of x/0 to be of infinite magnitude—which is, in fact, what the calculus textbooks attempt to teach. But as will be seen later, this is true only under limited conditions—one of which is that the dividend, x, also be of infinite magnitude.

Readers who are familiar with the surface of the Moebius strip (or the shape of Klein's bottle) shouldn't have too much difficulty in grasping the reasons underlying my theory that the quotient of x/0 = 0. (In fact, anyone who finishes reading the article may find that his mind has been bent into approximately the same shape.) The average computer programmer may find the idea easier to grasp if he associates it mentally with a wrap-around computer memory. And, for my mathematician friend, I will express the concept symbolically as follows:
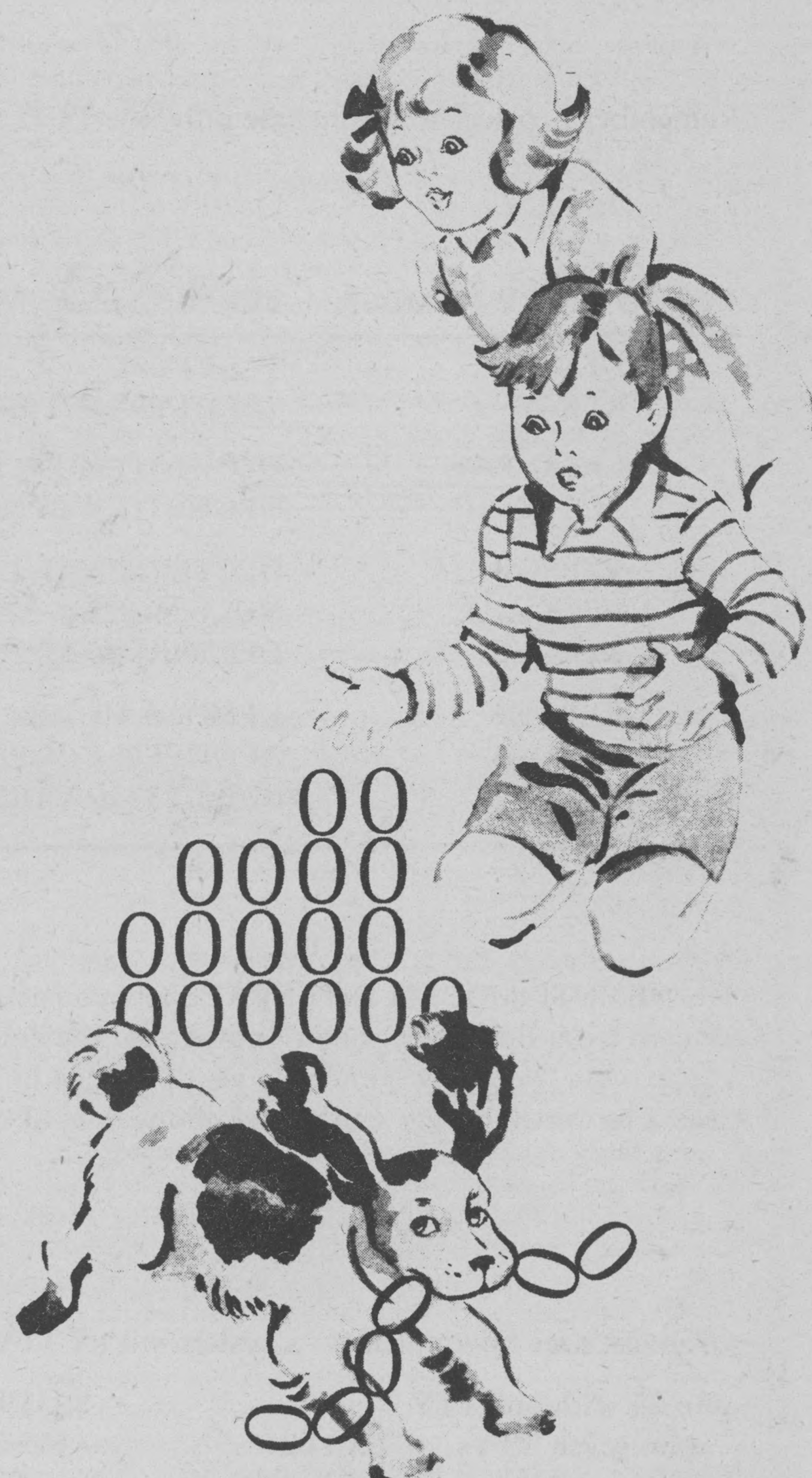
WRAP-AROUND INFINITY
(Also courtesy of Mr. L. Eisenzimmer)

$$\left| \frac{x}{a} \right| \longrightarrow \infty \text{ as } a \longrightarrow 0$$

$$\frac{x}{a} = 0 \text{ when } a = 0$$

But there is still a third way of approaching the problem of x/0, and I'm sure my mathematician won't overlook it. He may wish to interject, at this point, that when we divide x into y we are actually attempting to determine how many times we can subtract x from y without going negative. "Surely you can see," he will argue, "that we can subtract zero from any positive number an infinite number of times without going negative?"

Yes, I can see that. The trouble is, I can also see the fatuity of it. To my way of thinking, the question is not how many times we can subtract, but rather how many times we can subtract *successfully*. And I

submit that we do not add or subtract successfully unless we succeed in increasing or decreasing the original quantity. For example, when we add 5 to zero we have done something meaningful, because we have altered the original amount. But if we attempt to add zero to 5, we accomplish nothing. (We can alleviate the embarrassment of this dilemma by saying that we are adding zero *and* 5, rather than zero *to* 5.) The mathematician, however, adds zero to 5 with a flourish, smacks his lips in satisfaction, and deludes himself that he has obtained a constructive result. In actuality, he has merely gone through an exercise in futility, and obtained an *inevitable* result. If he has done anything constructive at all, it is to demonstrate the utter impossibility of adding zero to *anything*. Therefore, although we subtract zero from x an infinite number of times, we subtract successfully exactly *zero* times (the true quotient).

If my mathematician is still around, he will probably want to ask me how I propose to reconcile my original proposition (x, 0 = x) with the statement I just made (x/0 = 0). In order to bridge this seemingly impossible chasm, I must touch briefly on a subject which has gone too-long neglected—namely, the relativity of numbers. Obviously, numbers *are* relative, and the usual practice is to define them as either positive or negative with respect to zero. But we showed earlier that the word *nothing* can only be defined with respect to *something*, and the same is true here—i.e., zero itself can only be defined with respect to some other number (or numbers). If our hypothesis is correct that the true quotient of x/0 is zero, then the immediate problem is to isolate the *relative* value of zero on the imaginary mathematical scale (Cartesian horizontal axis). Since we know that zero lies exactly halfway between +n and −n, we can express the relative quotient of n/0 by the following equation:

$$\left(\frac{n}{0}\right)R = \frac{n - (-n)}{2} = \frac{n + n}{2} = \frac{2n}{2} = ñ \quad [1]$$

In other words, by halving the difference between +n and −n, we have found that the relative quotient of n/0 is a neutral n—i.e., it lies n units in a negative direction from +n, and n units in a positive direction from −n. (I have stressed the neutrality of n in this case by using the Spanish letter, ñ, which is doubly appropriate because the neutral n was discovered in New Mexico.) The practical application of this pragmadox is manifested in the fact that it satisfies the mathematician's craving for a unique result—i.e., it is not the same n that we would have obtained if we had divided n by 1 instead of by zero.

But my mathematician loves consistent results as well as unique results, and he won't overlook the apparent fact that my answer still doesn't check. He will be quick to point out that if my neutral n had a value, say, of 5, then 5 zeroes wouldn't make 5, and zero fives wouldn't make 5 either. Well, I absolutely agree that zero fives wouldn't make five, but I'm not so sure about the first proposition. If we start out with *one* zero, then multiply that zero by 5, it seems fairly reasonable to me that we should end up with *five* zeroes. In fact, I am gripped by an urge to place a string of five zeroes right here on the printed page, then ask my mathematician to count them for himself and *see* if they don't add up to 5. His immediate

response, naturally, would be: "Ah, but that is mere word-trickery. You are treating zeroes as if they were units, which isn't cricket at all." (Back to the old 0 = 1 pragmadox.)

Very well. For the time being, I'm prepared to let my mathematician have his way. We will treat zeroes strictly as non-units, and we will assume that there is no distinction in magnitude between 1 non-unit and 5 non-units. (To do otherwise would be to equate non-units with negative numbers.) Under these restrictions, I confess that my answer doesn't check. I can only say, by way of defense, that when my mathematician has a value, x, and doesn't divide it by anything (i.e., divides it by nothing), he is left with a value of x. And if then he divides that x by 1, he is *still* left with a value of x. But do *I* run around accusing *him* of saying that 1 is equal to *nothing*?!? It would seem that my neutral x, as a quotient for x/0, is valid for all practical purposes, since it is basically the same answer that my mathematician gets when he *doesn't* divide x by zero.

In any case, if x is the relative quotient of x/0, the true quotient may be expressed by taking the algebraic sum of +x and −x, then dividing by 2 in order to obtain the average:

$$\left(\frac{x}{0}\right)_T = \frac{x + (-x)}{2} = \frac{x - x}{2} = \frac{0}{2} = 0 \quad [2]$$

But here again my mathematician will attempt to pounce, tearing his hair and screaming that, in the first place, x/0 (can't be) = 0, because 0/x (is also) =0, and in the second place, how can x/0 be equal to x and zero at the same time (why don't I make up my mind, etc.), and in the third place, even if five zeroes do add up to 5, *zero* zeroes certainly wouldn't, because zero times *zero* is ZERO! (You know how these mathematicians always get in a lather about everything.)

Okay. In spite of the fact that this particular mathematician has been harrassing me ever since I began the article, I've grown somewhat attached to him. I think he is a good fellow at heart, and it gives me no great pleasure to stick another pin in his balloon. But I must gently point out that zero times zero, at least from a semantic point of view, does *not* equal zero. When we say that we have zero zeroes, we are actually saying that we have *no* non-units. And an absence of non-units implies the presence of an indefinite number of units. (In this case, my answer doesn't *exactly* check, but it doesn't exactly *not* check, either.)

My mathematician is not going to be happy about this at all. But please remember that we agreed to play the game according to his own rules. It was he who insisted that we treat zeroes as non-units. In fact, I think this conclusively proves that it is the mathematician who has furtively been treating zeroes as units.

And at long last we have reached the crux of the matter. The old nitty-gritty. The fallacy in the fallacy. Mathematicians have, for lo these many years, been harboring a mental image of zero as a non-unit, while simultaneously attempting to treat it as though it were a unit—a neutral unit, to be sure, but nevertheless as a unit. Well, we pays our nickel and we takes our choice. We are free to regard zero as a kind of neutral pseudo-unit, or we may treat it as a non-unit. But not both. If we elect to treat zeroes as non-units, we promptly deprive them of whatever neutrality they might have had, and they become essentially negative in character. (Hence the term *non*-unit or nothing.) Therefore, we cannot apply the same rules to a non-unit that we apply to true units, and expect the non-unit to meekly conform. As the mathematicians are fond of saying (or *were*, up to now), we simply cannot mix apples with oranges.

Now for a quick analytical summary of everything we've postulated.

1. If we treat zeroes as pseudo-units, then n/0 = n. (This is safe because, as previously noted, it is the same result that mathematicians get when they refuse to divide the number, n, by zero.)

2. If we treat zeroes as non-units, then n/0 = 0.

3. If we treat zeroes as pseudo-units, n × 0 = n. But we cannot mix pseudo-units with true units any easier than we can mix non-units with true units; therefore, to avoid confusion and stay on the safe side, we must express the product of zero and n as zero—with zero in this case being understood as representing n pseudo-units, distinguished from true units and non-units.

4. If we treat zeroes as non-units, then n × 0 = 0, provided n is not equal to zero; otherwise, the product is indeterminate.

5. The same reasoning applies when we divide zero by zero—i.e., the answer is 1 (necessarily expressed as zero) if we treat zeroes as pseudo-units, and indeterminate if we treat zeroes as non-units.

Conclusions: Plainly, we computer people are going to be in serious trouble if the mathematicians persist in regarding zeroes as non-units. We have already seen that multiplying one non-unit by another non-unit generates an indeterminate number of real units. There is nothing implausible about this, but it is equivalent to making something out of nothing, and we certainly don't want to be accused of that. Therefore, the only sane course of action is to treat zeroes as pseudo-units, whereby we common folk can more or less follow the conventional rules of mathematics.

Yes, that is the only path to follow, short of giving zero back to the Arabs; and I heartily recommend that we follow it.

(Unless, of course, there is a fallacy in the (fallacy in the fallacy).) ∎
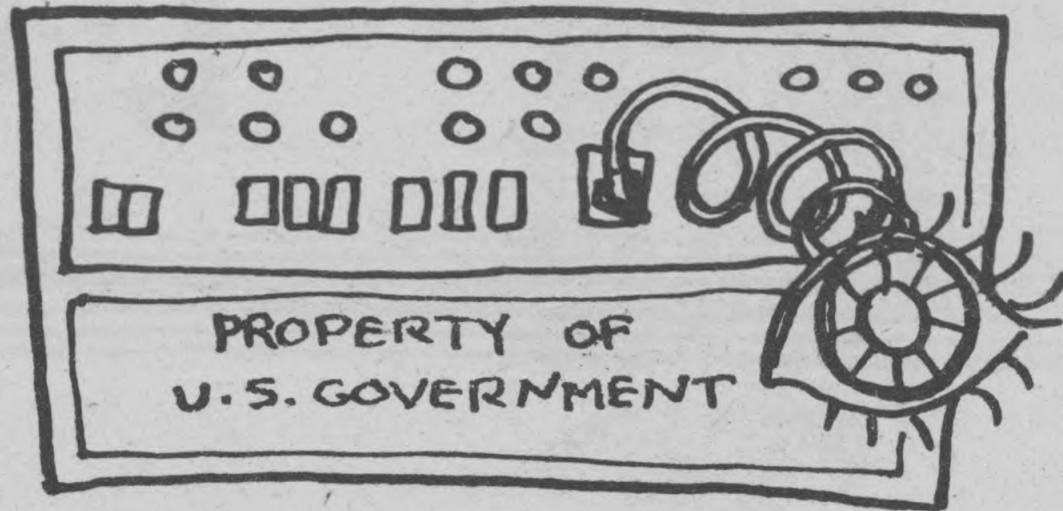
Books

Columbia Human Rights Law Review, Vol. 4, No. 1, Winter, 1972 Columbia School of Law, New York, N.Y. 301 pp. $4

A lawyer friend of mind tells me that I'm fortunate to have discovered this book in a marketplace glutted with turgid writing. For any computer professional with a social conscience, this particular issue at least, is an important publication. If you feel as I do that there may be alternatives to current institutional attitudes toward the increasing use of advanced technology—that is, it isn't all good just 'cause there's more—then this little volume will detail for you the many specific ways computers are being "used to abuse." One student author clearly summarizes that "an individual's integrity and his right to control information about himself is presently considered inferior to the profit motive."

The opening essays give a thorough philosophical insight into the politics of privacy. At the same time they deal thoroughly with the specific situations in which a rampant information technology has spotlighted the problems that occur when people prefer protection over privacy. Arthur Miller, author of *Assault on Privacy*, introduces the privacy environment with a sensitive appeal to "overcome our all-too-often complacent attitude toward the management of our affairs by what frequently are astigmatic administrators." Sen. Sam J. Ervin presents information from his "Hearings on Federal Data Banks, Computers and the Bill of Rights" that will chill your bones as well as it chills the First Amendment. Nicholas Katzenbach's dispassionate review of crime data centers is as humanistic as a JCL manual. ACLU attorney Frank Askin laboriously details the ways government surveillance in and of itself removes the spirit of participation from any concerned citizenry. He fought the Army and knows whereof he speaks. Michael Baker then tries to explain why the little guy can't protect himself from the misuse of information about himself.

If you think the picture is bleak by now, wait till the student authors get through detailing chapter and verse concerning "Political Surveillance," "Remote Camera Surveillance," and "Commercial Information Brokers." The thoroughness of their research is impressive. In summary, there is much here to provide some perspective on the rationale each of us uses to justify the long-term social cost of present practices in our particular areas of data processing.

Computers and Privacy—Arthur R. Miller: Writing in the winter of 1972, Miller incorrectly perceived (I admit I too was deceived) that "the political appeal of being for privacy is becoming apparent." The November election results showed that people want to be protected more than they want privacy. Miller's sensitive statement of the problem of the abuses of individual privacy in an increasingly complex society unfortunately holds little promise of correction. The result of enacting controls to control controls is that each new piece of legislation merely validates the belief that people cannot trust one another with personal freedom. To look for new legislation as an answer is a bit like the rational gambler who looks for a good strategy to take to Las Vegas. He knows that the best he can do is to lose slowly. If you believe that we are in "the only game in town," the gradual erosion of personal freedom should not be surprising.

First Amendment: The Computer Age—Sam J. Ervin, Jr.: Sen. Ervin knows his business. One could not ask for a more thorough evaluation of the uses of computers by government as a tool to control "free speech." In IBM Sales School, we were told we could ask the visiting V.P. any questions we wished at the upcoming Friday night bull session. If you were "wise," you were careful to ask only "nice" questions. Sen. Ervin really cares about freedom of speech and doesn't want individuals to feel intimidated by their government. I'm afraid sales school tactics are a lot more practical than legislation. Information technology is clearly too powerful to control, and one's only chance for survival may be to simply to tell them what they want to hear (popularly known as "shucking"). How else can an applicant combat (for one example from the many Ervin details) the administrator Ervin investigated who protested that "only the computer" analyzed the answers to compulsory questions like the following:

"I believe there is a God."

"I believe in the second coming of Christ."

"My sex life is satisfactory."

"I wish I were not bothered by thought about sex."

Surely there need be no moral stigma to shucking any computer inquiry that you as an individual consider immaterial.

Crime Data Centers—N. Katzenbach and R. Tomc: Katzenbach started antitrust action against IBM while he worked for Kennedy. Now he's IBM's chief counsel. He exhibits an elitist audacity by writing "White collar crime has grown increasingly sophisticated in the areas of securities, thefts, embezzlement and political corruption" and then suggesting that his company's tools can protect us from petty larceny. Ralph Nader is far better equipped to reply to the kind of outrage, for example, inspired in me by the authors' conclusion: "A debate cast in terms of improved law enforcement *versus* privacy is pretty sterile once the emotion is dissipated." I don't believe that dissipating emotion is of value unless, of course, one seeks to emulate a computer.

Social Science and Surveillance—Frank Askin: The author's conclusions are right on. He fought the Army intelligence program and knows about the incredible arrogance of the military. If you want all the background to show that "people today are indeed chilled in their political participation, in their speech, in their associations, and in their right to petition for redress of grievances," he provides it. Unfortunately, he's hopelessly wordy and takes paragraphs to amplify his feeling with such statements as: "Jerome Bruner has shown that past experiences, needs, and expectations influence perception . . ." If you don't know that surveillance in and of itself is a bummer, you probably won't get it out of a bunch of authoritative logic trips. However, it's all here if you want support.

Record Privacy Consciousness—Michael Baker: A coauthor with Alan Westin on *Data Banks in a Free Society*, Baker points out in this essay that the little guy cannot keep from getting worked over by most systems—school, welfare, credit, police, etc. Since shucking won't do him much good now because he's involuntarily in these systems, his only hope must come from outside—but Baker won't say where. (Perhaps he means Nader, Common Cause, ACLU, etc.) He gives a neat story about welfare workers laughing at an old lady turning in a rumpled piece of paper with her age written on it as a substitute for lost immigration papers. You are then led to infer that at the same time as her welfare request is being denied, some middle-class citizen is paying an inappropriately high interest rate on his home based on "facts" stored on a rumpled disc file.

Change of consciousness can occur when you discover "it's" happening to you, too.

The following three articles are by students who are not given name credit. I wonder why?

Political Surveillance: It's hard for the layman to judge whether consideration of Fourth Amendment guarantees in addition to First Amendment safeguards would serve to protect individuals from malicious political surveillance. I find it hard to believe that any government agency would be deterred from putting a tail on someone simply because they might have to pay taxpayers' money as damages for violations of illegal search. In spirit I'm with the author—it's too easy for the government to protect itself as it tends "to approach these dissident domestic organizations in the same fashion it deals with unfriendly foreign powers." From a computer standpoint, the key word in this article is "dataveillance." This term describes a process of correlating data from several files to determine possible "need for surveillance." Help!! My experience is that you're lucky if any one file can yield an accurate picture of what it was intended to describe. If security analysts were to appreciate the complexity of a mere aged accounts receivable application, they would not be so enthusiastic in their haste to infer that some John Doe is potentially disloyal.

Remote Camera Surveillance: This article describes an example of a super-advanced technology being used in crime prevention. The results of the 24-hour tv camera system on two blocks of Mt. Vernon, N.Y., are admittedly unimpressive, but the expectations are tremendous. "To begin, police can use a Mt. Vernon-type surveillance system to read a pedestrian's lips or to read documents in his possession . . . to observe and magnify people in their apartments, cars, or on the street . . . to enforce dormant or inconsistently utilized statutes; e.g., jaywalking, spitting, or failing to license your dog." Referring to *Privacy* by Professor Fried, the author points out that "once a person has complete surveillance of someone, that surveillance extinguishes any possibility of trust." And so it is with citizens and their government.

Commercial Information Brokers: This very comprehensive review details in case after case how a difference in scale can become a difference in kind. Some misuses of information data banks are merely distasteful, such as junk mail (if you can be nonchalant about the charge that "$240 million of the $321 million deficit sustained due to third class mail is a result of commercial mailings"). Others "of these intrusions would have been impossible if the information was manually handled and manually disseminated." The following example is selected from the hundreds of abuses the author documents that occur with mailing lists, income tax information, employment information and census data, motor vehicle registrations, and many other tax-supported functions. The Veterans Administration had furnished data tapes to several organizations—Red Cross, American Legion, VFW, DAV; but refused to furnish the same tapes to the Vietnam Veterans Against the War. The VA was finally ordered to yield the VVAW this information. The court ruled that the earlier release of the data required the additional release. The VA said they would not have released the data in the first place had it not been on computer tape. In conclusion, the author notes that the Freedom of Information Act and Fair Credit Reporting Act are ineffective as methods of redressing wrongs already committed.

I still believe that complete openness is the best defense against invasion of privacy. But our fear of one another will apparently be paramount for some time to come, and so I too must urge restraint on the use of computer data banks. If we become more inclined to not underestimate the goodwill of others, those who seek to protect us for our own good may become less inclined to ask the computer to engage in provinces of the human domain.

—Lynn Stoller

THE PROGRAMMER'S

by marc le brun

[Each issue we will present an "advanced" programming technique: with explanations, examples, programs and problems. We welcome suggestions for topics of interest to you.]

TOOLBOX

POINTERS

Data comes in different flavors. The two types of data that BASIC users are most familiar with are numbers and strings. Many BASIC systems also provide the user with a third data type; matrices. Occasionally there are other types as well, such as logical values, which are either zero or one, and various peculiar types such as are often associated with files. There is an important type of data however, which is not available on any BASIC system – the *pointer*. In this article we will discuss how to implement pointers in BASIC.

The pointer is the fundamental unit of *structural information*. Structures are so universal in computing that it is impossible to even begin to examine them in any detail in this article; the techniques presented here should be used as a basis upon which you can build more sophisticated programming practices.

Pointers are values which are used to indicate *relations* (or "connections") between other units of data. To create structures from this data each of the items in the structure must have a unique value associated with it, called it's location or *address*. For example, suppose that we wish to connect ten values stored in a one dimensional array in some sort of a structure. Then the integers 1, 2, . . . , 10 would be the addresses of those values. If the ten values were stored in a 2 by 5 array then the ordered pairs (1,1), (1,2), . . . , (1,5), (2,1), (2,2), . . . , (2,5) would be the addresses of those values, and so on. For the sake of simplicity, we will assume that the addresses we will be discussing can be represented as single numbers.

A pointer is simply a variable whose value is an address. Since we have assumed that the addresses we are talking about are single numbers we can use ordinary BASIC variables to represent pointers. Pointers can do only one thing; they can connect or "point at" data items. Pointers are *not* ordinary numbers; in particular adding, subtracting or doing other sorts of arithmetic with pointers has no meaning. Dividing a pointer by three, for example, makes as much sense as dividing a string by three. Pointers *just* point, and I think I have made my point . . . .

To avoid confusion with ordinary numbers pointers are often represented by arrows when they appear in diagrams. Let's call our array of ten values A, and let's say that the variable P is a pointer which is pointing at the sixth element in A. We would diagram it like this –
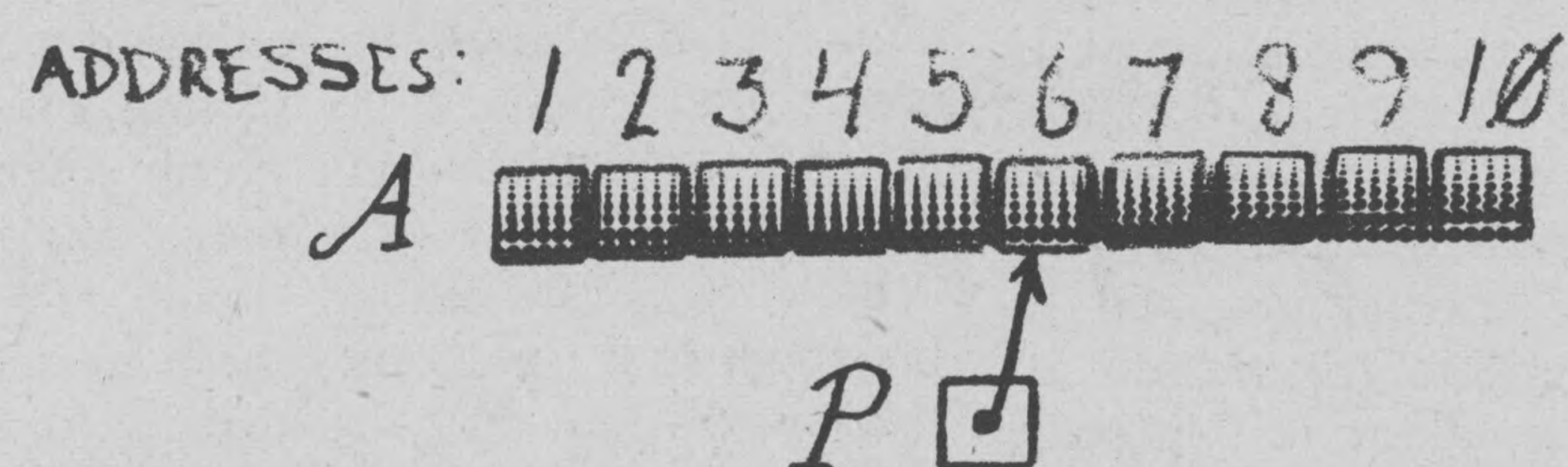

The program on this page creates a list of N random numbers sorted in increasing order. It does this *without* moving the random numbers in the array. Data items consist of one number and one pointer. The pointer points to the *next largest* random number in the list. The largest number has a null pointer associated with it. Initially there is a special item at the N + 1 address containing a zero (smallest possible number) and a null pointer. When a random number is generated it is stored in A. Then the program searches down the list *following the pointers*. When it finds a number larger than the one just generated, it inserts the current number into the list prior to that item. If the search reaches the end of the list it attaches the new element to the end. Notice that these insertions are accomplished simply by changing the values of the pointers. Study this program until you understand how it works.

INSERTING AN ITEM BY ALTERING POINTERS.
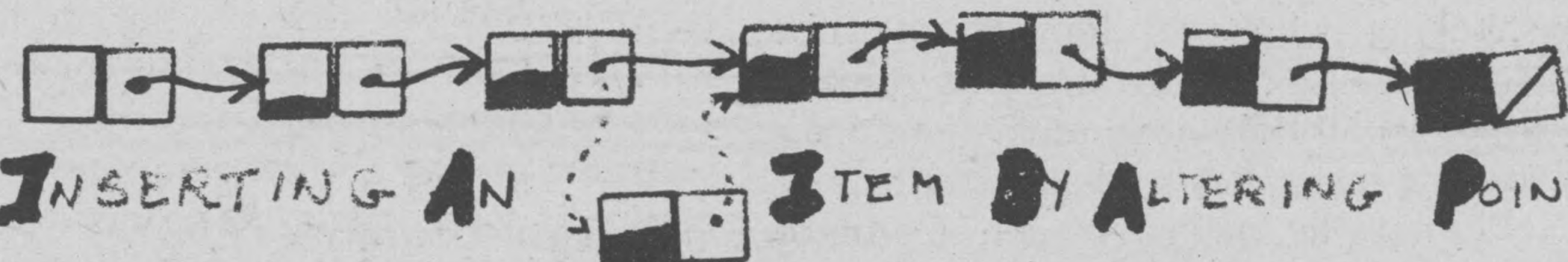
HOW WOULD YOU "PROGRAM" A MAP OF THE USA? LOOK AT PAGE 4. HOW WOULD YOU WRITE A CAVES PROGRAM? HOW WOULD YOU PROGRAM" THE EDGES OF A POLYHEDRON? HOW WOULD YOU PROGRAM THE STRUCTURE OF A SENTENCE? OR A ROAD MAP? OR A FAMILY TREE? OR . . . ?
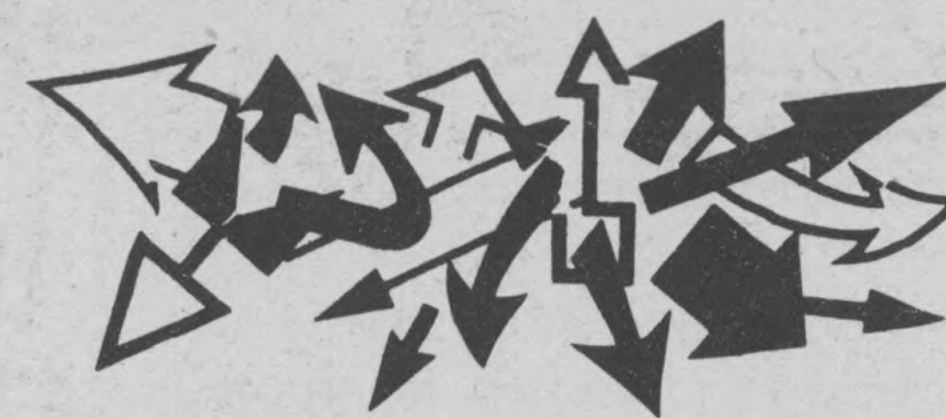
There is a special pointer value that is often very useful. It is called the *null pointer*. It is a pointer that doesn't point anywhere. In this article we will use the number zero to represent the null pointer. The null pointer is also often called NIL (usually in capital letters) with reference to the LISP language, which has contributed a number of other weird words to computer science, and which was one of the first computer languages to deal specifically with structures.

In diagrams, NIL is often represented as a diagonal slash through the box representing the pointer, with the slash in the same orientation as the slash through the numeral Ø (you do slash your zeroes don't you?!).

Since pointers are easily represented by numbers they may be stored in arrays, possibly in the same array as the rest of the data. For instance, in the example below, each complete row of the array A is considered a data item, the first element in the row is a number and the second elements in the row are pointers to other rows.

Now you know about pointers. Here is the example. And remember the best things in life are sometimes the simplest.

ORDER OUT OF CHAOS
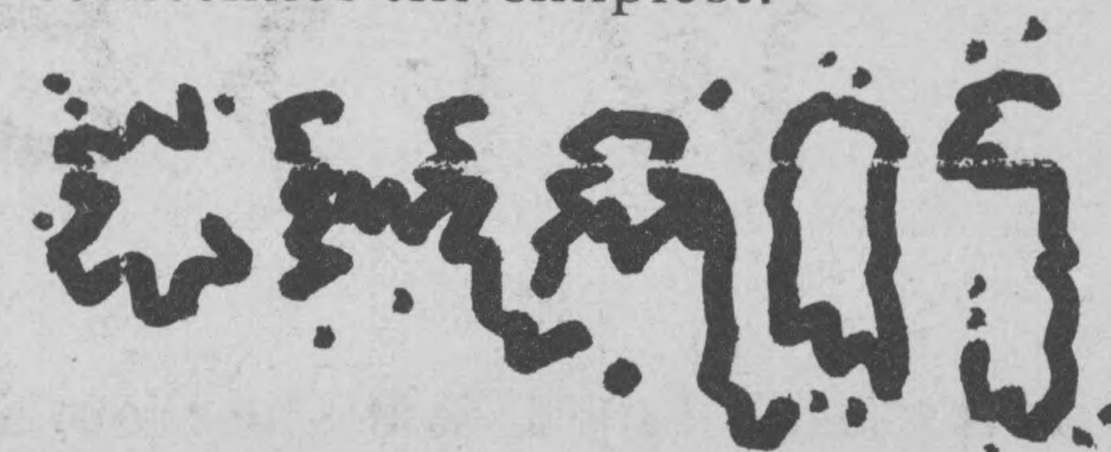
```
10   DIM A[100,2]
100  REM *** TALK TO HUMAN ***
110  PRINT "HELLO, HOW MANY VALUES ";
120  INPUT N
130  PRINT
140  A[N+1,1]=0
150  A[N+1,2]=0
200  REM *** GENERATE RANDOM VALUES ***
210  FOR I=1 TO N
220  A[I,1]=RND(0)
230  J=N+1
300  REM *** LOOP TO FIND PLACE TO INSERT ***
310  K=A[J,2]
320  IF K=0 THEN 400
330  IF A[K,1]>A[I,1] THEN 400
340  J=K
350  GOTO 300
400  REM *** INSERT IN LIST ***
410  A[I,2]=K
420  A[J,2]=I
430  NEXT I
500  REM *** PRINT THE ARRAY ***
510  PRINT "ORIGINAL ARRAY :"
520  PRINT
530  PRINT "ADDRESS","VALUE","POINTER"
540  PRINT
550  FOR I=1 TO N
560  PRINT I,A[I,1],A[I,2]
570  NEXT I
580  PRINT
600  REM *** PRINT THE LIST ***
610  PRINT
620  PRINT "SORTED LIST :"
630  PRINT
640  PRINT "ADDRESS","VALUE","POINTER"
650  PRINT
660  J=N+1
700  REM *** LOOP TO PRINT LIST ***
710  K=A[J,2]
720  IF K=0 THEN 800
730  PRINT K,A[K,1],A[K,2]
740  J=K
750  GOTO 700
800  REM *** SAY BYE BYE ***
810  PRINT
820  PRINT "BYE BYE HUMAN !!!"
830  PRINT
840  END
```

DON'T BE A DONUT... run it!

22

TELETYPES

Cheap Tape Winder

FINGER POWER!

cardboard
teletype spindle
slit
pencil
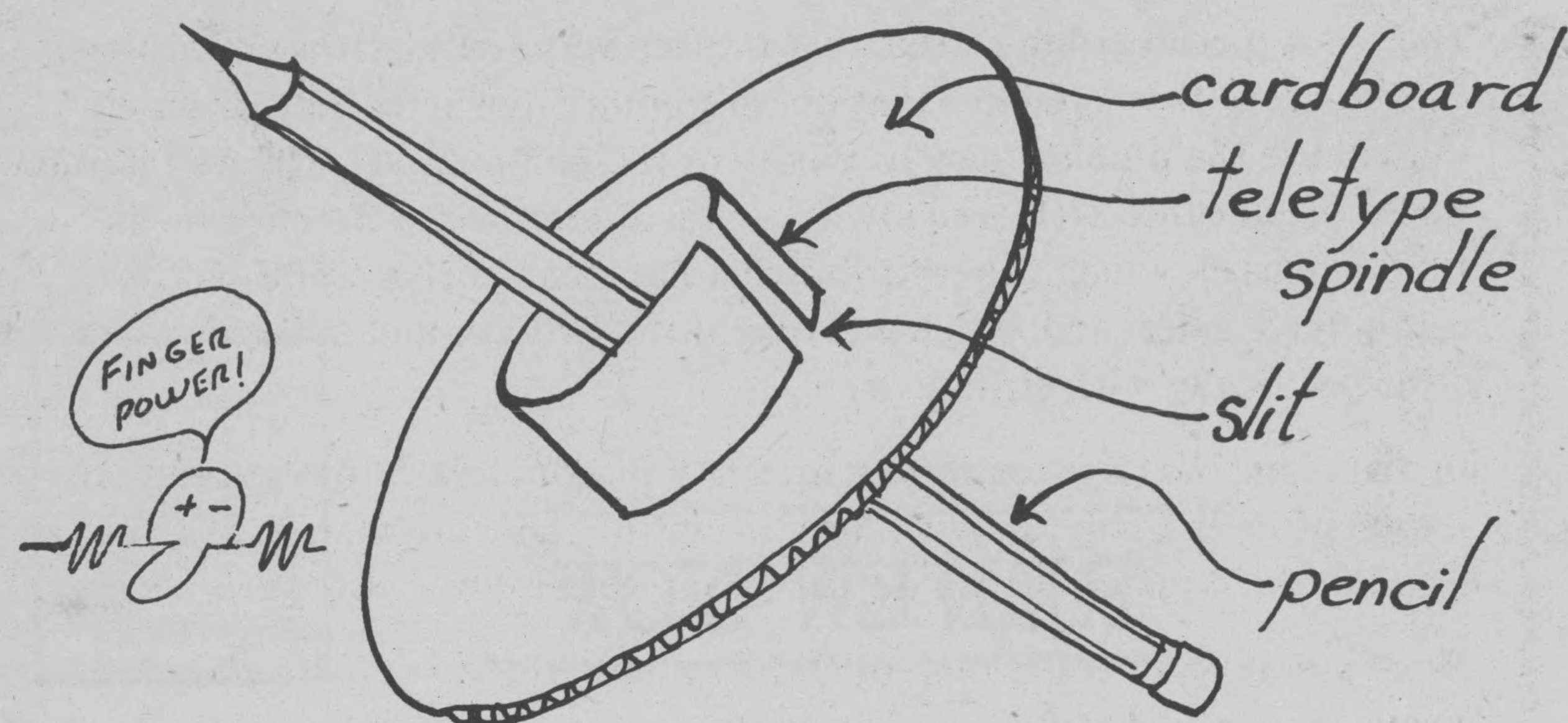
WHERE TO PUT A ROLL OF PAPER TAPE WHILE IT'S BEING READ BY THE TTY READER

WI$E?

Is it wise to buy a Teletype from Teletype Corp?

In the interest of saving money we checked out the possibility of buying teletypewriters from Teletype Corporation directly, rather than buying through a local dealer (which is what we previously recommended).

We found Teletype Corp. to be most cooperative. They suggested that interested readers contact their Los Angeles sales office at (213)724-6040 or their central number (312)982-2500, or write directly to the plant:

Teletype Corporation
Sales Department
5555 West Touhy Ave.
Skokie, Ill. 60076

You should note the purchase of a Teletype is a cash operation, payment is due in total in 30 days. If you want to lease a TTY, you'll have to deal with RCA, Western Union or a local dealer. You should also note that prices are F.O.B. plant, which means the buyer pays the freight charges from Skokie, Illinois — this will probably run a minimum of $20 per unit.

We were unable to get any definitive response to a question on delivery dates. Rumor has it that you will wait 6—8 months for delivery when you buy from the factory. Even if delivery is only 4 months (we've been wrong before) you have to PLAN AHEAD.

Once delivered, your TTY will have to be "installed." Though it's checked out at the factory, 1000 miles in a truck may beat it up. Installation (final test and adjustment) by Teletype will cost you $75/unit plus mileage if you're more than 25 miles from an office. Our friends at Data Terminals will only charge about $35/unit if you have the machine shipped to their office in San Jose — then they will deliver it to you. Data Terminals claims that unless you have an experienced man at your school to check out your new TTY, you could possible find yourself in *real* trouble a few weeks later.

Well, is it worth it? We priced out the following unit which is perfect for a T/S terminal. ASR 33 Data Terminal with paper tape reader/punch, friction feed paper, automatic reader control (X-on, X-off), ME type wheel (give a slashed zero), pedestal mount including chad box and copy holder.

Model 3320/5JC		
To which we add	Price	$970
	Transportation	20
	Installation	50 (happy medium)
	TOTAL	$1040 (coupler is not included)

The same unit from our friends at Data Terminals is only $1100 and they can deliver quickly and will assume all responsibility for bad units etc.

Is it worth it? We don't think so. Not if you're buying onesy, twosy units. The $60/unit profit you give to your local dealer will save you more than that in aggravation and risk. BUT, if you're buying lotsa terminals, it might well be worth the risk to buy all your units direct from Teletype and have them installed by your local men.

Other words of wisdom —

You can buy a builtin modem from Teletype also. The modem runs an additional $200 or so. It must be hard wired to a telephone Data Access Arrangement (DAA). You cannot use a builtin acoustic coupler with the 3320/5JC for some strange reason. You have to to use a separate coupler sitting on the floor or nearby.

Teletype recommends maintenance every 750 hours or 6 months. How do you keep track? For $15 more, they will equip your TTY with an elapsed timer that records the "motor-on" time to the nearest hour. We think it's a worthwhile investment! Model SOP 188660.

There are a zillion other options available when you're buying a TTY. The unit and information described above is basic, but will meet your needs adequately.

REMEMBER — If you plan to hardwire your TTY to a DEC or Data General computer, you need a modification kit which may run you more than $190 installed.

Paper

In our first issue we recommended PERFECTION brand paper (stock number 6210) which we were buying as cheap as $0.73 per roll. We haven't been able to beat that price but we found another good paper you might check out. We buy it from Blake Moffitt & Towne paper company in San Francisco. We buy the 4½" diameter roll, white, model 40-18303 for $0.82 per roll if you buy 60 rolls, $0.77 per roll if you buy 120 rolls. Higher quality bond paper, model 41-18303, will run you $1.15 per roll in 60 roll lots and $1.08 in lots of 120.

HAND DANDY TTY WHEELER AROUNDER

We got our from Penney's ($13.99 semi-deluxe model, they had some for $6.99).

LeRoy

23